2002年2月1日発行（毎月1回1日発行）第21巻第2号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

FEB. 2002

2

定価
550Yen

掲載プログラムがダウンロードできる！ http://www.basicmagazine.net/

集中連載 第2弾 20点確保!! 大学入試センター試験対策

数学①②「BASIC言語」予想問題と解法

特選投稿プログラム・コーナー

- Visual Basic V5.0 Cut!
- Visual Basic V6.0 迷路
- Visual Basic V6.0 キートリック
- Visual Basic V6.0 速食
- VBScript+DirectX 風流
- VBScript+DirectX スピード
- Visual C++ V6.0 BCJOYクラス＆サンプル
- C++Builder4 BALLてーじ
- F-BASIC V6.3 影さんのパソ破ッ!（十六武蔵）
- HSP V2.5 神経衰弱 あわせてポン
- HSP V2.5 ルパン
- iアプリ 移植版 L-Quest for i

CPUの歴史と未来をさぐる！
21世紀も現役として稼働！8ビットCPU"Z80"の巻

感染してもあわてない!!
コンピュータ・ウイルス「BADTRANS.B」撃退法

パソコンアップグレード実践講座
旧型FMVをWinXPマシンに復活！

誰でもマクロがマスターできる
マイクロソフトExcelでプログラミング

プログラムの内容を磁気でワイヤレス送信
ナビゲータ搭載の思考型自走ロボット「INSECT」

編集部の学校訪問記

ホームページやゲーム制作にがんばるパソコン部
東京都江東区立深川第三中学校

8Mフルレート
進化した2つのブロード

新時代のブロードバンドルータに必要だった

8MフルレートADSLに対応する高スループットエンジン搭載。
さらに多彩なPC接続インタフェースと充実機能を実現。
これこそ先進の快速・快適ワイヤレスブロードバンドルータ。

- ●IEEE802.11b準拠の高速ワイヤレスLANを実現。
- ●ADSLをはじめCATVにも対応する余裕の高スループット。
- ●WEBブラウザで行うカンタン接続設定。
- ●さらに進化した「らくらくアシスタント」で設定もらくらく。
- ●ダイヤルアップでも利用できるマルチライン機能。
- ●ネットワークゲームなどで必要なポート設定を簡単切替えのAPプロファイル機能。
- ●セキュリティにも注力した実力のルータ機能。

・PPPoE（フレッツ・ADSL）対応・DHCPサーバ機能・DHCPクライアント機能・アドバンスドNAT・ダイナミックポートコントロール機能・不正アクセス検出機能・ログ機能・パケットフィルタリングなど

*軽量・高速スキャン・強力駆除！100日間試用の無償アップデート付Windows®用ウイルス対策ソフト「ウイルス警備隊V3」を添付CDに収録。

■高スループットエンジン搭載ワイヤレスブロードバンドルータ

Aterm WBR75H ワイヤレスLANベース NEW

Aterm WBR75H ＋ 本体装置用 WL11CA（1枚）

WAN／100BASE-TX：1　RS-232C：1（ダイヤルアップ用）
LAN ／IEEE802.11b、100BASE-TXスイッチングHUB：4、USB：1

■お得にはじめるなら **Aterm WBR75H ワイヤレスLANセット（カードタイプ）** NEW

Aterm WBR75H ＋ 本体装置用&PC接続用 WL11CA（各1枚）

■ブロードバンドルータ **Aterm WBR75H** NEW

■ETHERNETボックス **Aterm WL11E** NEW

Windows®以外のPCやゲーム機でもワイヤレスを実現。さらにWAPモード搭載によりワイヤレスの親機にも変身の1台2役。　*設定時はWindows®搭載PCが必要です。

■128bitWEP対応ワイヤレスLANカード **Aterm WL11CA** NEW

■ワイヤレスLANカード **Aterm WL11C** ■USBボックス **Aterm WL11U(W)**

*ワイヤレスLANのネットワーク上にWL11C/11U（W）が加わった時はWEP(40bit)での対応となります。

! フレッツ・ADSL / Yahoo! BBなどブロードバンドサービス各社に対応。
事前にホームページ「AtermStation」記載の[接続確認済ブロードバンド事業者リスト]をご確認ください。

マルチライン機能搭載。設定もらくらく、ブロードバンドを
気軽に手軽にはじめるなら。— Aterm WARPSTAR Σシリーズ

- ◎1.5M　ADSLモデム内蔵ブロードバンドルータ　**Aterm WB65DSL**
- ◎ISDN&ブロードバンドルータ　**Aterm WB55TL**
- ◎ブロードバンドルータ　**Aterm WB45RL**

*各機種とも、カードタイプ、USBタイプとのおトクなワイヤレスLANセットがあります。*すべてオープン価格。

WBR75H　WL11E

Aterm® WARPSTAR

Microsoft® Windows®XP 対応　掲載の新商品はWindows®XPに対応しています。

■ご使用の際は必ず「取扱説明書」をよくお読みの上、正しくお使いください。■Aterm®は日本電気（株）の登録商標です。■Windows®は米国マイクロソフト社の登録商標です。■ランプ表示はハメコミ合成です。また、ランプ表示の内容は利用イメージのため、実際の動作と同一ではありません。

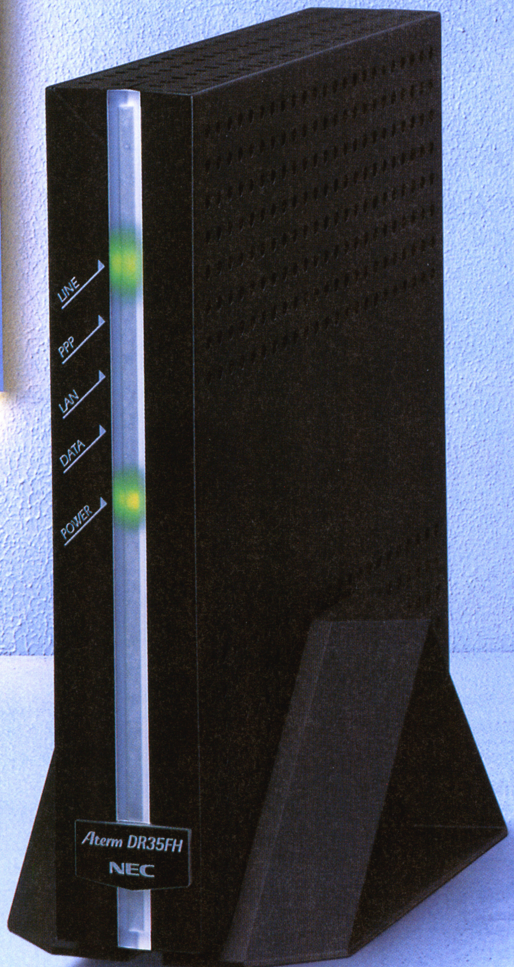
ユーザ登録キャンペーン実施中！
2001 11/26（月）〜 2002 3/31（日）

詳しくは「AtermStation」ホームページ http://aterm.cplaza.ne.jp/

Empowered by Innovation NEC

ADSLに対応。
バンドルータ⊿〈デルタ〉誕生。

のは、このスピード、高機能、そして快適性でした。



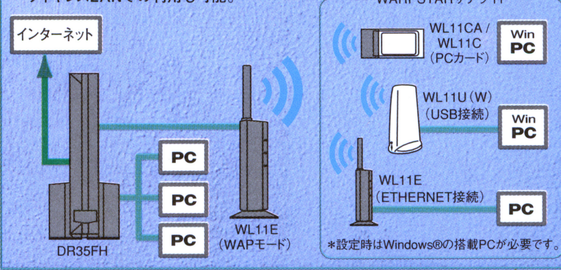
DR35FH

8MフルレートADSL**モデム内蔵の本格派ブロードバンドルータ。これ1台で、高度なネットワーク利用にもフレキシブルに対応する、まさに本格派ADSLルータ。**

- 「8Mbps（G.dmt）／1.5Mbps（G.lite） AnnexC方式」対応のADSLモデムを内蔵。
  *8Mbpsサービスでの接続事業者はアッカ・ネットワークスまたは、同等のサービスを提供する事業者です。
- 高度な設定も柔軟にカスタマイズできる高機能ブロードバンドルータ機能。
- 外部公開用サーバを他と切り離しファイアウォールのDMZ領域化も可能。
- ビジネスユースにも対応する、豊富なルータ機能を満載。

・スタティックルーティング（最大40経路設定可能）・DHCPサーバ機能・ProxyDNS機能・IPパケットフィルタリング機能（IPアドレス/Macアドレス等による指定可）・IPマスカレード機能・ポートマッピング（ポート無変換機能付）・システムログ機能など
*DIRECTSTAR⊿も「ウイルス警備隊V3」を添付CDに収録。

■WL11EをWAP（ワイヤレスアクセスポイント）として接続すれば、ワイヤレスLANでの利用も可能。



■4ポートスイッチングHUB搭載フルレート対応ADSLルータ
Aterm DR35FH/GS NEW
WAN／ADSL回線接続:1　LAN／100BASE-TX/10BASE-T:4
ACCA NETWORKS 推奨

■フルレート対応ADSLルータ
Aterm DR30F/GS NEW
WAN／ADSL回線接続:1　LAN／10BASE-T:1
ACCA NETWORKS 推奨

※本商品でご利用いただけるADSL事業者およびインターネットプロバイダをホームページ「AtermStation」で必ずご確認ください。

掲載商品はすべてオープン価格です。

Aterm® DIRECTSTAR⊿

●「121ware.com」ホームページ http://121ware.com

商品に関するお問い合わせは

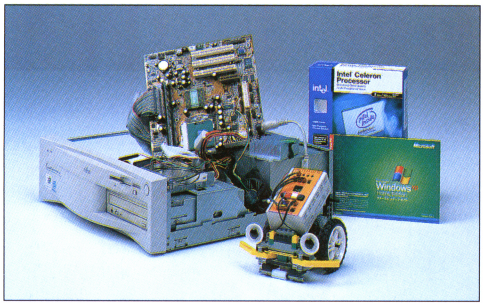
●Aterm（エーターム）インフォメーションセンター 0120-361138 TEL.0471-85-4761
受付時間:月～金曜日 午前9時～午後6時、土曜日 午前9時～午後5時（日曜日、祝祭日、年末年始、当社の休日はお休みさせていただきます。）

NECアクセステクニカ

マイコンBASIC Magazine

http://www.basicmagazine.net

2 2002 FEBRUARY



©マイコンBASICマガジン2月号
第21巻 第2号 2002年1月25日印刷
2002年2月1日発行
定価 550円（本体524円）
電波新聞社 2002 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号 141-8715
電話 東京(03)3445-6111(大代表)
振替 (東京)00150-3-51961
印刷：奥村印刷(株)　製本：(株)堅省堂
<本誌掲載の記事,プログラムの無断使用を禁止します>

BASIC Magazine FEBRUARY 2002 CONTENTS

巻頭カラー企画

（編集部の学校訪問記）ホームページやゲーム制作にがんばるパソコン部
東京都江東区立深川第三中学校 8

突撃潜入レポート
インテル冬のPCスクール 10

集中連載

20点確保!! 大学入試センター試験対策直前マスター
数学①②「BASIC言語」の予想問題と解法 第2弾 11

特選投稿プログラム・コーナー 35

Visual Basic V5.0	Cut!	木村雄一	36
Visual Basic V6.0	迷路	戸田文彰	41
Visual Basic V6.0	キートリック	山下慎平	46
Visual Basic V6.0	速食	はりまお	48
VBScript+DirectX	風流	黒田国男	50
VBScript+DirectX	スピード	金井義男	52
Visual C++ V6.0	BCJOYクラス＆サンプル	BlueCube	54
C++Builder4	BALLてーじ	赤羊旅団	59
F-BASIC V6.3	影さんのパソ破ッ！（十六武蔵）	ロディマス	65
HSP V2.5	神経衰弱 あわせてポン	三好利治	68
HSP V2.5	ルパン	くろすけ	73
iアプリ	移植版　L-Quest for i	小倉邦治	76

今月のベスト・プログラマーは誰の手に！
明日のスタープログラマー 79

今月のベスト・プログラムを解説！
Dr.Dのベスト・プログラム研究室 80

感染してもあわてない!!
コンピュータ・ウイルス「BADTRANS.B」撃退法 17

CPUの歴史と未来をさぐる！
21世紀も現役として稼働！8ビットCPU“Z80” 22

パソコンアップグレード実践講座
旧型モデルをWindowsXPマシンとして復活！FMV DESKPOWERのマザーボード／CPU交換 27

マンガで学ぶ！コンピュータの世界
母なるマザーボード!? …… 82

これから始めるブロードバンド活用術
ADSL体験記　～基礎講座～ …… 86

PDAで自作ソフトを持ち歩く！
ポケットに入るコンパクトサイズ PC用携帯ゲーム機「PIECE」プログラミング …… 91

もっと身近にプログラミング
チャレンジ！オリジナルプログラム言語の制作 …… 95

フリーソフトでプログラミング
HSP完全マスター　3Dプログラミング …… 99

図解！Visual Basicゲーム作成術
キャラクター・アクション・ゲーム⑤ プレイヤー・キャラクターの逆襲 …… 104

はじめよう　Microsoft ExcelでBASICプログラミング
第10回 コンピュータ対戦型のパズルゲーム～オセロゲーム 109

プログラムの内容を磁気でワイヤレス送信
ナビゲータ搭載の思考型自走ロボット「INSECT」のすべて 115

21世紀の遊びはコレ！ ロボ道場
「全国高専ロボコン」レポート＆階段を降りるASIMO／二足歩行ロボット格闘競技大会情報 …… 120

読者のコーナー
あくせすROOM／プレゼント …… 124

ソフト・メーカーのPRのページ
SOFTWARE MAKER HOT INFORMATION …… 125

広告索引

ハード		**パソコンショップ**	
NEC	表2～前付カラー3	若松通商	記事中34
ソニー	表4	HIT&RUN	後付128
レック制御	記事中119	HRDサービス	後付128
ソフト		ガレージコム	後付129
日本ファルコム	表3	ASERA SYSTEM	後付129
学校		シグナル	後付129
日本ゲームソフト学院	後付127	パシフィックネット	後付130

好評発売中!!

楽しく学習できる
新まんが電気学入門

監修：江田　稔
（文部科学省初等中等教育局
漫画：原子　力
●A5判／247頁　送料 380円
定価1,500円（税込）

電気のはたらきから
パソコンのしくみまで
わかりやすくまんがで解説！

主な内容

●第１章／静電気
静電気（まさつ電気）／正の電気と負の電気／はく（箔）検電器／静電気と電流／雷とは？……他

●第2章／乾電池と豆電球
電池／電池の原理／乾電池のしくみ／豆電球のしくみ／電気を通すもの、通さないもの／導線のつなぎ方／配線図と回路記号／豆電球の直列つなぎ……他

●第3章／電流と電圧
閉回路と開回路／電流／電荷／電圧／抵抗（電気抵抗）／抵抗のいろいろ／抵抗値の見分け方／オームの法則／電流計（直流用）電圧計（直流用）のしくみ……他

●第4章／回路計算
電圧の総和／電流の総和／電圧のかかり方／抵抗の直列接続と合成抵抗値／合成抵抗の中の未知抵抗／内部抵抗の求め方／キルヒホッフの（第１）法則……他

●第5章／電気エネルギー
電流による発熱作用／電流による発熱原理／発熱量の測定／ジュールの法則／抵抗の接続方法と発熱量の大きさ・電力／電力量／積算電力計（交流用）……他

●第6章／磁石と電磁石
磁石と磁極／Ｎ極とＳ極／磁力の大きさ／磁気誘導／Ｎ極とＳ極の分離／磁力線／地磁気／電流の大きさと方位磁針の振れ／右ねじの法則……他

●第7章／電磁誘導
電磁誘導／レンツの法則（誘導電流の向き）／ファラデーの電磁誘導の法則／コイルの巻き数と誘導電圧／電気ブランコによる電磁誘導……他

●第8章／電流と電子
火花放電／真空放電とグロー放電／蛍光灯と電球／クルックス管と陰極線／原子と電子／電子と火花放電／トランジスタの構造……他

●第9章／パソコン
パソコンって何？／パソコンでどんなことができるの？／パソコンはどう動くの？／パソコンのいろいろな形態／パソコンの簡単な歴史／パソコンの仕組み―CPU・メモリ・補助記憶装置・フロッピーディスク・ハードディスク・CD-ROM／周辺機器―キーボード・マウス・ディスプレイ・プリンタ・スキャナ・デジタルカメラ（デジカメ）／OSの働き―OSって何？／OSの働き―OSの種類は？／日本語入力ソフト／アプリケーションソフト／LAN（ラン）て何？／USBとは？／パソコン関連の単位／インターネットとは？／インターネットはどうしてできたか？／インターネットサービスプロバイダ／インターネットで何ができるか？／WWWとは？／電子メール

電波新聞社
東京本社　電話（03）3445-8201（ダイヤルイン）
〒141-8715 東京都品川区東五反田 1-11-15 電波ビル

マイコンソフト

XVGA-1PROとは

XVGA-1PROは、DOS/Vパソコン（AT互換機）をはじめ、NEC PC-9800シリーズ、Apple PowerMacintoshの画面出力を、家庭用テレビで見ることができるように映像信号を変換する装置です。

大型テレビや、液晶プロジェクターなどに接続して、パソコンを利用したプレゼンテーションをする際に適しています。

本機は、多くの映像端子出力をそなえており、ビデオ端子（※）、Sビデオ端子（※）、HD/コンポーネント・ビデオ端子（※）、D端子（※）を持つ家庭用テレビや映像機器に接続し、パソコン映像を表示することが出来ます。

本機で変換された映像は、NTSCだけではなく、ハイビジョンやVGA（水平同期周波数31kHz）の映像としても出力できるため、お客様がハイビジョン対応テレビや、パソコン映像入力対応テレビをお持ちの場合は、NTSCより鮮明な映像を表示することができます。

（※）各端子の形状はカタログ裏面をご覧下さい。

高画質TVは、より鮮明な映像出力コンバーターを選ぶ
ニュータイプD端子／HD端子対応ダウンスキャンコンバーター登場！
こんなコンバーターが欲しかった!!

オン・スクリーン・メニューとリモコン操作も充実

変換した映像の表示位置や明るさなどは、テレビ画面に表示された項目を選択して調整できます。また、離れた位置から操作ができる『リモコン』やXVGA-1 PRO本体の設置場所によってリモコン受光部が隠れてしまっても、リモコンを使用できるように『リモコン受光ユニット』も付属されています。

使って便利な画面ズーム機能

画面の拡大機能を持つスキャンコンバーターは、珍しくありませんが、XVGA-1PROの拡大機能は、より使いやすくなるように工夫されています。

9等分されたテレビ画面がリモコンに割り当てられ、拡大したい位置を直感的に選ぶことができる『エリアズーム機能』。リモコン上の3つのボタンに拡大したい位置と拡大率を登録できる『ホーム位置登録機能』などがあります。

拡大した画面は、拡大率を変更したり、上，下，左，右，斜め，などの方向に表示位置を移動させることができます。

ワイドテレビの画面比率もバッチリ!

4：3のテレビと16：9のワイドテレビでは画面比率は異なります。通常のテレビ放送もスキャンコンバーターで変換した映像もワイドテレビで表示すると、横長になってしまいますがXVGA-1PROではこれらの問題を解決するため、『WIDE_TV』機能が付いています。この機能を使用すると、ワイドテレビでも画面比率1：1の映像を表示することができます。

新しいパソコンからPC-9801まで対応!

最近のDOS/V（AT互換機）パソコンやApple Power Macintoshはもちろん、PC-9801シリーズの24kHz（640×400ドット）にも対応しています。

24kHzのRGB信号をNTSC信号に変換する機能は、他のコンバーターでもできますが、本機は24kHz（640×400ドット）を31kHz（640×480ドット）にアップスキャンすることができます。これにより、入手が困難になった24kHz対応モニタの変わりに現在のDOS/V用モニタが代用できることになります。また、XGA（1024×768ドット）やSVGA（800×600ドット）の画面をVGA（640×480ドット相当）に変換して表示する機能を備えていますので、VGA画面（640×480ドット）しか表示できないパソコン対応テレビやVGAプロジェクターにもXGA画面やSVGA画面を表示することができます。

対応機種と対応画面モード

対応機種	対応画面モード	水平同期（垂直同期）
NEC PC-9801シリーズ	640×400ドット	24kHz（56Hz）
各社DOS/Vパソコン（AT互換機） または NEC PC-9801/9821 NEC PC-98NX	640×400ドット	31kHz（70Hz）
	640×480 ドット	31kHz（60Hz）
		37kHz（72Hz）
	800×600ドット	35kHz（56Hz）
		37kHz（60Hz）
		48kHz（72Hz）
		46kHz（75Hz）
	1024×768ドット	48kHz（60Hz）
		56kHz（70Hz）
		60kHz（75Hz）
Apple Power Macintosh ※4	640×480ドット	35kHz（67Hz）
	800×600ドット	37kHz（60Hz）
	832×624ドット	49kHz（75Hz）
	1024×768ドット	48kHz（60Hz）
		61kHz（75Hz）

※4 MacintoshはApple Computer社の登録商標です。

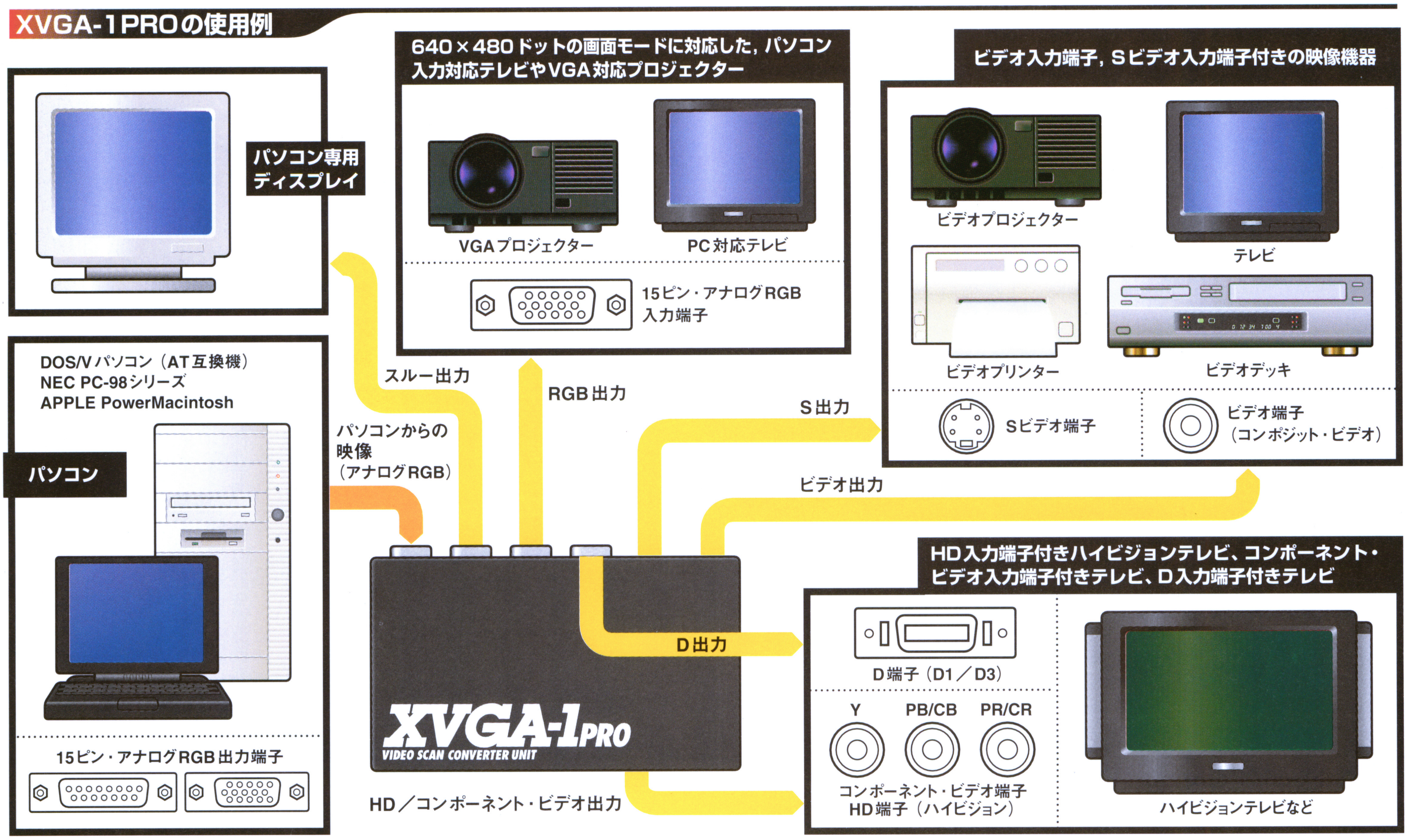
ハイビジョン・テレビ対応ビデオ・スキャンコンバーター・ユニット

好評発売中

XVGA-1PRO
VIDEO SCAN CONVERTER UNIT

標準価格56,800円（税別）

JANコード 4961997001066 DP3913360



ご購入前の注意事項

全般の注意事項

- 図1、または図2（※5）のアナログRGB出力端子を持たないパソコンには接続できません。
  本機をご購入前に、お客様のパソコンに図1または図2（※2）のアナログRGB出力端子（映像出力端子）があることをご確認下さい。
  ディスプレイ一体型のデスクトップ・パソコンや、デジタル方式の液晶ディスプレイが接続されている機種などでは、アナログRGBの映像出力端子を持たないことがあります。
  ※5　図2の端子はゲームポート（ジョイスティック端子）ではありません。
- 対応表に載っていない画面モードの映像は、本機で変換することができません。
- ノートパソコンでは、本体にアナログRGB端子が無くても、アダプター（ポートリプリケーター）などでRGB映像を出力できることがあります。
- パソコンのデジタル液晶ディスプレイ用端子であるDVI端子やDFP端子には接続できません。
- 本機のスルー出力映像は、パソコンに直接ディスプレイを接続した時よりも若干鮮明度が低下します。あらかじめご了承ください。
- NTSC、VGA、ハイビジョンの映像を同時に出力することはできません。
- XVGA-1PROとパソコンを接続する際、RGBセレクター（ディスプレイ切替機）やRGB分配機を介して接続すると、正しく映像を変換できないことがあります。
- デジタル液晶ディスプレイ端子（DFP端子など）とアナログRGB端子の両方の端子を持つ機種では、これらの端子を同時に使用できない機種があります。これはパソコン側の仕様であり、この様な機種では、デジタル液晶ディスプレイとXVGA-1PROの映像を同時に表示する事ができません。（FUJITSU FMV Cシリーズなど）
- XVGA-1PROは、パソコンを使用したプレゼンテーションやデモンストレーション等には適していますが、色合いや画面比率の違いから、テレビに映った画面を見ながらの画像データの作成（CG）やCAD設計等には適しておりません。
- ノートパソコンでは、外部ディスプレイ（ここではXVGA-1PROのこと）に映像を出力すると、内蔵の液晶ディスプレイを表示できない機種があります。これはパソコン側の仕様です。
- 複合同期信号（Csync）のみに対応したディスプレイを本機に接続した場合、そのディスプレイには映像を映すことができません。
- インターレス方式の映像には対応しておりません。
- パソコンの1024×768ドットを越える画面サイズ（1280×1024ドットなど）の映像には対応しておりません。
- 1024×600ドットや852×480ドットなどのワイド画面モードを持つパソコンの映像は、正しく変換できないことがあります。

NEC PC-98シリーズ接続時の注意事項

- アナログRGB出力端子を持たない機種や端子の仕様が異なる機種には接続できません。
  以下の機種は接続できない機種の例です。
  PC-98FineNXシリーズ（FN23C,FN20Cなど）
  PC-9821　V10/C4R、V7/C4K、Cu10、
  Fineシリーズ（F200,F166）、Ts、Cbシリーズ
  PC-9801 NV、NL、NL/R、PC-9801CVなど
  H98シリーズ
  PC-9801,U,M,Fなど、デジタルRGB出力しか持たない機種
- 以下のPC-9800シリーズのノート型は、接続の際「CRT接続ケーブル（NEC　PC-9801NS/L-01）」が必要です。「CRT接続ケーブル」はNEC社より発売されています。別途、ご購入ください。
  PC-9801NS/L、NS/A、NL/A
  PC-9821シリーズ（※6）
  ※6 ポート・バーが付属されているノート型のPC-9821シリーズに接続する　際は「CRT接続ケーブル」は必要ありません。
- 以下のPC-9801のノート型は、接続の際「CRTパック（NEC　PC-9801　NS/E-14U）」が必要です。「CRTパック」はNEC社より発売されています。別途、ご購入ください。
  PC-9801NS/E、NS/T、NC、NS/R、NX、NX/C、NA、NA/C
- PC-98NXのノート型で、アナログRGB出力端子（図1の端子）の無い機種では、「ポート・バー」が必要です。
- PC-98NX　LBシリーズ、PC-98NX　VA20Cは映像出力端子が特殊なため、別売の「マウス/ディスプレイケーブル（NEC PC-VP-BK01）」が必要です。

Apple PowerMacintosh接続時の注意事項

- PowerMacintosh以前に発売された機種（旧Macintoshシリーズや Macintosh II シリーズ）への接続はMacintosh側に図2の端子があれば接続可能ですが、一部映像信号が特殊な物もあるため、完全な動作保証をすることができません。あらかじめご了解の上、本機の購入をご検討下さい。
  基本的にCPUにPowerPCを使用している機種が本機の対応機種となりますので、ご購入時の参考にしてください。
  以下の機種は当社が動作を保証できる機種です
  Power Macintosh 6000, 7000, 8000, 9000, G3, G4各シリーズ
  PowerBook 5300, 2400, 3400, G3の各機種
- ディスプレイを接続する端子（ビデオポート）を持たないMacintoshには接続できません。本機を購入される前に、Macintosh本体をご覧いただき、ディスプレイを接続する端子（ビデオポート図1または図2の端子）があることをご確認ください。
  以下の機種には映像出力端子が無いため接続できません
  iMac, iBook, Performa5000シリーズ, PowerBook1400シリーズなど
- Power Book5300シリーズに接続の際は、PowerBook付属の「ビデオ出力ケーブル」または「ビデオアダプタケーブル」が必要です。
- 本機が対応する映像出力は、Macintoshに標準で搭載されているビデオカード（またはオンボード）の映像出力のみです。サードパーティなどから販売されているビデオカードやお客様が増設されたビデオカード（特に旧型の物）は、一部特殊な信号を出力している機種もありますので、動作保証はいたしかねます。
- PowerBookなどのPCMCIA（PCカードスロット）に接続するビデオカードでの動作保証はいたしかねます。

テレビとの接続について

- ビデオ入力端子やSビデオ入力端子の無いテレビには接続できません。
- 解像度の低いテレビなどでは、文字などが極端に読みづらくなります。あらかじめご了承ください。
- NTSC方式のテレビにビデオ接続ケーブル（コンポジット・ビデオ）で接続すると、テレビ側のY/C分離回路の性能によって、画質や鮮明度が極端に低下することがあります。テレビにSビデオ入力端子がある時は、Sビデオ・ケーブルなどで接続する事をお薦めします。
- ハイビジョンテレビであっても、D3端子またはHD入力端子（外部MUSEデコーダーの映像を色差信号で入力する端子）が無い機種では、ハイビジョンの画質で映像を映すことができません。HD入力端子（Y, PB/CB, PR/CR）またはD3端子の無いハイビジョンテレビでは、ビデオ端子やSビデオ端子での接続になります。また、本機はハイビジョンのチューナー端子（MUSE端子など）に接続することはできません。
- 本機のコンポーネント出力端子（Y,PB/CB,PR/CR）から出力される映像信号は、「Y,PB/CB,PR/CR」または「Y,PB,PR」の映像入力端子を持つテレビに接続可能です。「Y,CB,CR」としか表記されていない端子は映像信号の仕様が異なるため接続しても映像を正しく表示できません。
- プログレッシブ表示対応テレビやDRC表示対応テレビにビデオケーブルやSビデオ・ケーブルで接続すると、本来の鮮明度を得られなかったり、画像の細部がうまく表示されないことがあります。
- 本機はD2（525p）の映像を出力することはできません。
- ビデオ・プロジェクターや、プロジェクション・テレビは、その特性上ブラウン管方式のテレビに比べ、十分な鮮明度が得られず、文字などが極端に読みづらくなります。ご了解の上、本製品をご購入ください。
- プロジェクション・テレビは特性上ブラウン管方式のテレビに比べ、画面の焼き付きが起こりやすいため、ご注意ください。
  また、通常のブラウン管方式のテレビや液晶テレビであっても、長時間同じ映像を表示し続けると画面の焼き付きの原因となりますのでご注意ください。

ご購入後の注意事項

- もしお客様が、テレビ画面の点滅や激しい色の変化を見ると気分が悪くなるような体質をお持ちの場合は、本機のご使用をご遠慮ください。大画面のテレビでご使用の際は、特に注意が必要です。
  また、ご使用の際は部屋を明るくし、テレビからできるだけ離れて画面を見てください。
- 本機で変換した映像をVTRなどで録画する際、画像や映像に著作権が存在する時は、録画する画像や映像（ソフトウェアも含む）の著作者（版権元）の了解が必要になることがあります。
- 本機で変換した映像をVTRなどで録画し、その映像を販売・複製・貸与・放送・上映する行為は、著作権等を侵害する恐れがありますので注意が必要です。

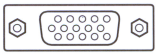
図1

アナログRGB端子（15ピン3段, HD-SUB）

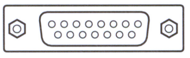
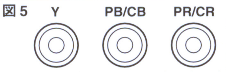
図2

アナログRGB端子（15ピン2段, D-SUB）

図3

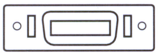
ビデオ端子（コンポジット・ビデオ）

図4

Sビデオ端子

図5


コンポーネント・ビデオ端子HD端子（ハイビジョン）

図6

D端子（D1／D3）

発売元：**電波新聞社　販売部**
〒141-8735 東京都品川区東五反田1-11-15
☎03-3445-8201（ダイヤルイン）

開発・企画：**マイコンソフト株式会社**
〒530-0005 大阪市北区中之島3-2-4 朝日新聞ビル6F
（電波新聞大阪本社内）　☎06-6203-2827

マイコンソフト社ホームページ
WEB
http://www.dempa.co.jp/micomsoft/index.html

東京本社	03-3445-6111（大代表）	札幌支局	011-641-5591	関東総局	027-326-3206	金沢支局	076-263-8661	高松支局	087-861-3111
大阪本社	06-6203-3361（大代表）	仙台支局	022-227-7211	松本支局	0263-36-2266	京都支局	075-221-8021	鹿児島支局	099-226-3630
西部本社	092-431-7411（大代表）	新潟支局	025-245-2526	名古屋支局	052-261-4541	広島支局	082-228-5581		

▲パソコン部のみなさん

編集部の学校訪問記

ホームページやゲーム制作にがんばるパソコン部

東京の下町・深川は江戸情緒が残された街です。今回は，深川第三中学校のパソコン部におじゃましました。

同校は平成１１年・１２年度の江東区教育委員会インターネット活用調査研究協力校として，「インターネットを利用した授業の研究ー総合的な学習を見据えてー」と題した研究を行ってきました。現在，その成果が授業や部活動などで存分に発揮されています。

東京都・江東区立深川第三中学校

http://www.koto.ed.jp/fuka3-chu/

着実に課題を克服！平成12年8月，新機種導入でインターネットも快適

「インターネットを利用することで授業がどのように変わるのか」「情報発信能力を養うことが重要になる」という考えから，江東区教育委員会インターネット活用調査研究協力校として研究に取り組んだのが，平成11年。

当時使用していたPCがNECの「PC-9821Cx2」（MS-DOS＆Windows3.1）ということもあり，「インターネットの同時アクセスやプリントアウトに対応しきれず，授業どころではなかった」と当時を振り返るのは，研究担当の宮本先生。

「ただ，トラブルが続出するなかでも，パソコンの利用状況は良好で，試行錯誤の中，生徒の学習効果が高まることは創造できました」とも。

現在使用しているPCは，平成12年8月に導入された富士通の「FMV6566SL5c」（Windows98）。パソコン室は技術科，社会科などを中心にした授業，パソコン部が主に使用しています。「先生方や地域の方たちが使用するケースも増えてくるでしょうから，使用に関してのルール，管理は徹底したいですね」とは研究主任の田内先生。

パソコン室の設備

- **サーバ**
  富士通PRIMERGY ES200（Windows2000）
- **クライアント**
  富士通FMV6566SL5c（Windows98）…20台生徒用
  富士通FMV6600DX4e（Windows98）…1台 先生用
- **インターネットは「東京ベイネットワーク」のCATV**

昨年，「総合的な学習の時間」で行われたさまざまな授業

インターネットバーチャルショップ体験

田内先生が指導のもと，生活や契約をテーマにした「電子商取引」を疑似体験する公開授業が行われました。

電子商取引の問題点や信用できる業者選択の能力を身につけることで，自己選択・責任の大切さを学ぼうというもの。インターネットを利用することで，授業の幅が広がりました。

コンピュータソフトウェア著作権協会（ACCS）の著作権啓蒙授業

全校生徒620人を対象に，ACCS主催の「クリエーターになる子どもたちへ」と題した，著作権保護に関する授業も行われました。著作権保護ビデオの鑑賞，著作権○×クイズなどを通して，著作権の大切さを学びました。

これをきっかけとして，ACCSでは，同様の出張授業を順次展開しています。

▲コンピュータ教育に理解のある壷内明校長先生

▲インターネット利用研究主任の社会科・田内寛人先生。同校にかかせない存在

▲同校のホームページ

▼パソコン室には「FMV6566SL5c」が20台生徒用として設置されている

▲OSはWindows98

◀パソコン部の活動は，各人それぞれが思い思いのゲームやホームページを作成。和気あいあいながらも，文化祭展示を目標に真剣

元気なパソコン部

同校のパソコン部には現在，3年生が9人，2年生が7人，1年生が6人の計22名の部員が在籍しています。主に火曜日と金曜日が活動日となっていて，パソコン室に全員集まってきます。

普段の活動内容は，「HTML」でのホームページ制作，フリーソフト「HSP」やゲーム作成ツール「クリック＆クリエイト」などを使ったゲーム制作です。部員それぞれが自由に好きなことをやっている中にも，教えあったり，情報交換しあったりと，自身の能力向上にがんばっています。

活動の大きな目標は年1回秋に開かれる“文化祭”への作品出展です。例年，パソコン部制作のゲームを，一般の方たちに体験してもらっています。

昨年は名刺作りなども試み，大変好評だったようです。

パソコン部部員紹介

▶パソコン部顧問で数学科の宮本俊行先生

部長
金沢　昭くん
(3年生)
パソコンは奥が深く楽しいですね。部をまとめながらがんばっています。

副部長
桐生充啓くん
(3年生)
パソコンは小学校の頃から使っていました。主に絵を描いていたんですよ。

田村勇樹くん
(3年生)
小学校でロゴライターを使っていました。プログラムは楽しいですね。

横尾拓也くん（3年生）
小学校でホームページを作ったのがきっかけで，インターネットに関心を持ちました。

寺津和彦くん
(2年生)
気分で入部したというのが本音です。ただ小学校でパソコンに触れたことが興味の始まりです。

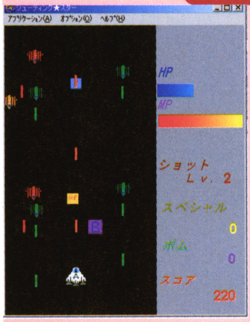
澤口尚玄くん
(1年生)
小学生のときインターネットをはじめました。パソコンやプログラムについてもっと知りたいですね。

昨年の文化祭で展示されたゲームの数々

「はいぱ～☆シューティング」

エンドレス型シューティング・ゲーム。通常ショットや魔法弾，必殺技を使い分け，敵を倒していく。

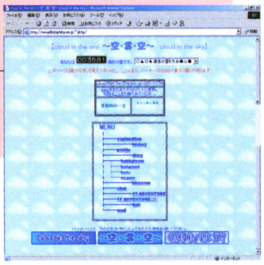
「シューティング★スター」



ジャンルはシューティング・ゲーム。エンドレスで高得点をねらうというゲーム。

金沢君のホームページ「cloud in the sky ～空・雲・空～ cloud in the sky」からダウンロードして，試すことができます。

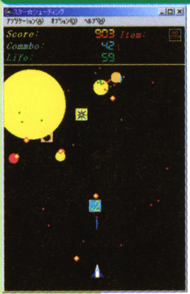
http://www5b.biglobe.ne.jp/~akhp/index.html



「トト♪の冒険2」

トト♪君を操作して，地下に落下しながら宝石を集める。針に当たるなどするとライフが減る。アクション・ゲーム。（※編集部内でも熱中してプレイしていました）

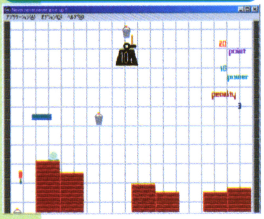
「スター☆シューティング」



敵をまとめて倒す快感が楽しい非スクロール・シューティング・ゲーム。アイテムを使い分け，ハイスコアを狙う。

「バトルブロック崩し」

「Never.Never.Never give up!!」



これらのゲームは，田村くんのホームページ「どり～むスター」からダウンロードして試すことができます。※インターネットランキングも実施中です。ぜひ，挑戦してみましょう！

http://www5.ocn.ne.jp/~yuukihp/

◀取材時は，技術科の伴先生指導のもと，江東区に提出するという，ホームページ紹介の展示物を作っていた

東京都江東区立深川第三中学校 〒135-0044 東京都江東区越中島3-7-1 ☎03-3641-5948

突撃潜入レポート インテル冬のPCスクール

パソコンを使ってオリジナル・クリスマス・ツリーを作成，テクノロジーで創造性を拡大しよう

毎回趣向を凝らしたテーマで行われるインテルのPCスクール。この冬のテーマは「3D作成ソフトを使って，クリスマス・ツリーのオーナメント作りに挑戦しよう！」というもの。抽選で選ばれた東京近郊の小学校4年生～6年生19人が参加しました。

午前中にパソコンやインターネットの基礎知識を学んだ子どもたちは，お昼をはさみ，検索サイトを利用して，自力でクリスマスやオーナメントについて調べました。続いて，3Dソフト「六角大王」の使い方を学びながら，それぞれ思い思いの形のオーナメントを作成。展開図作成ソフトを通したデータをプリントし，手作業で作品を組み立て上げました。

完成したオーナメントは，インテル東京本社のある国際ビルに設置された約3メートルのクリスマス・ツリーに，子どもたちの手で飾り付けられました。

◀▲難しい理屈を覚えるよりも，まず実際に見たり触ったりすることから。CPUも手にとって，じっくり観察

▲「キーボードファイター2」を使って，ゲームしながらタイピングの練習

▲検索サイトを利用してクリスマスについて調べ学習

▲3Dって何だろう？　立体的に見える原理について学んだあと，実際に3D作成ソフトを使って立体画像作りを体験

▲いよいよオーナメント作り開始！　星型や雪だるまなど複雑な形にチャレンジする子もいれば，長方体のプレゼントボックスですましちゃう子までいろいろでした

▲展開図作成ソフトで開かれた図形を厚手の用紙にプリント。あとは，ひたすら切って，折って，貼るべし！

▲モールやシールで飾り付けたら完成！

▲みんなそろって「メリー・クリスマス！」

Not expensive, but expansive!

※Intel、Intel Insideロゴ、Pentiumはアメリカ合衆国およびその他の国における
インテルコーポレーションまたはその子会社の商標または登録商標です。

TSUKUMO
www.tsukumo.co.jp

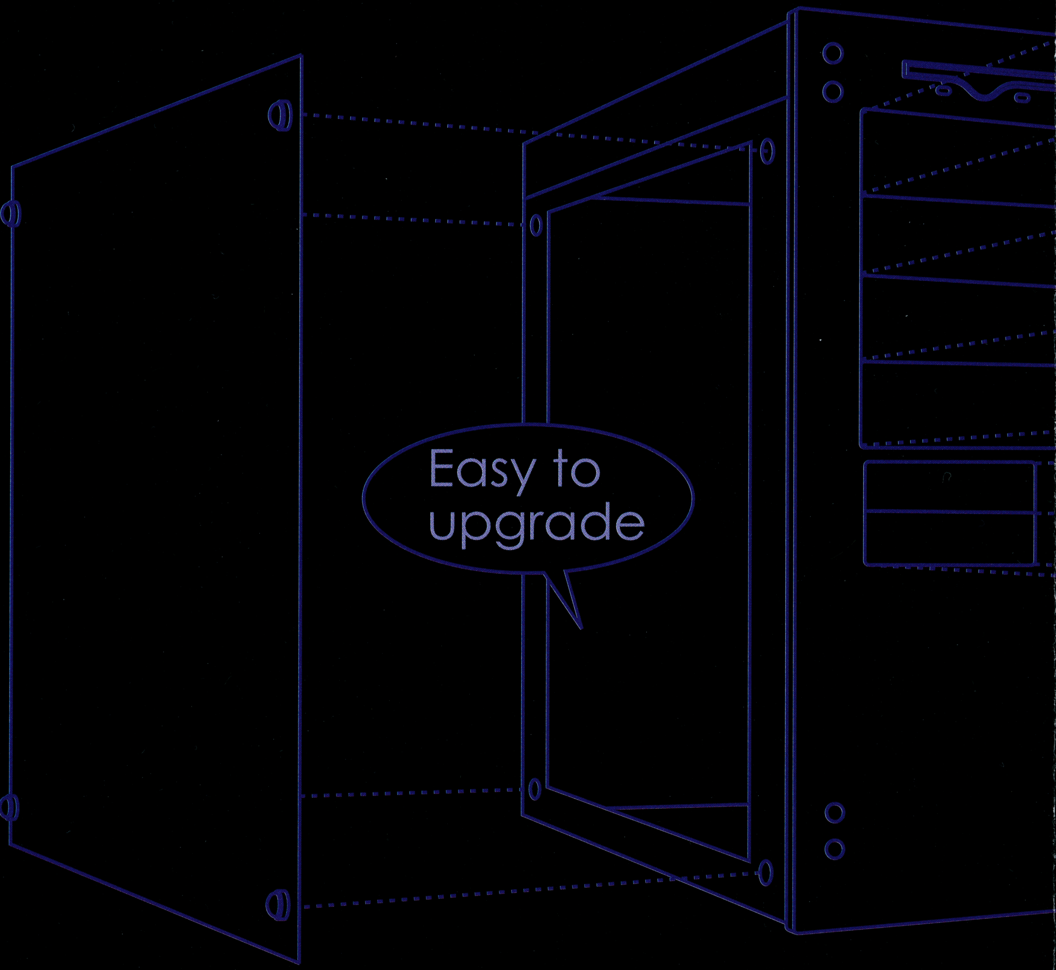
インテル® Pentium® 4プロセッサ搭載「TS-JADE」は、優れたパフォーマンスを提供します。

Intel® Pentium® 4 プロセッサ搭載

ツクモオリジナルPC

jade

TS-Series



最新のIntel® Pentium® 4プロセッサ 2.00GHz搭載

インテル® Pentium® 4プロセッサ搭載「TS-JADE」を。ひろがるデジタルワールドの中心に。

TS-JADE2000登場！

詳しくは裏面をご覧下さい。

TSUKUMO
www.tsukumo.co.jp

」は、優れたパフォーマンスを提供します。

インテル® Pentium® 4プロセッサ搭載「TS-JADE」を。
ひろがるデジタルワールドの中心に。

intel inside pentium® 4

TS-JADE 1600

JustLink/JustSpeed搭載 COMBO DRIVEモデル

●CPU:インテル®Pentium®4プロセッサ 1.60GHz
●マザーボード:インテル®845チップセット搭載
●メインメモリ:256MB(PC133 CL3/最大:3GB)
●ビデオカード:nVIDIA®GeForce2 MX™ 400/64MB (SDRAM)
●ハードディスク:40GB(UltraATA100/5400rpm)
●COMBO DRIVE:書込12倍速、書換10倍速、読込32倍速、DVD読込8倍速
B's Recorder GOLD & B's CLIP & Power DVD VR-X添付
●フロッピー:2モード
●ブロードバンド:10BASE-T/100BASE-TX
●サウンド:AC'97準拠(オンボード)
●キーボード:日本語109キー(PS/2)
●マウス:スクロールマウス(PS/2)
●ケース:W210×D470×H400mm(AL-588/330USB)
●電源:330W

Windows XP Homeモデル
TS-1600/40ACXP ………… ¥115,999

Windows 2000 Professionalモデル
TS-1600/40ACS ………… ¥125,999

ツクモオリジナルPC
jade
TS-Series

よくばり仕様の
お買得モデル

●オリジナルアルミケース(AL-588/330USB)

Intel® Pentium® 4プロセッサ 1.60GHz搭載

TS-JADE L1600

JustLink/JustSpeed搭載 CD-R/RW DRIVEモデル

●CPU:インテル®Pentium ®4プロセッサ 1.60GHz
●マザーボード:インテル®845チップセット搭載
●メインメモリ:256MB(PC133 CL3/最大:3GB)
●ビデオカード:nVIDIA®GeForce 2 MX™ 200/64MB(SDRAM)
●ハードディスク:40GB(UltraATA100/5400rpm)
●CD-R/RW DRIVE:書込20倍速、書換10倍速、読込40倍速、
B's Recorder Gold & B's CLIP添付
●フロッピー:2モード
●ブロードバンド:10BASE-T/100BASE-TX
●サウンド:AC'97準拠(オンボード)
●キーボード:日本語112キー(PS/2)
●マウス:スクロールマウス(PS/2)
●ケース:W325×D470×H105mm(H340D)
●電源:180W

Windows XP Homeモデル
TS-L1600/40CRXP ………… ¥99,999

Windows 2000 Professionalモデル
TS-L1600/40CRS ………… ¥109,999

●省スペースながらも
パワー十分

10万円を切った
普及モデル

●スリムタイプケース(H340D)

【ツクモ法人営業部】

■東京地区〈秋葉原〉TEL. 03-3253-3199 FAX. 03-5295-8741
〈八重洲〉TEL. 03-5201-5699 FAX. 03-5201-5698
■名古屋地区 TEL. 052-561-1990 FAX. 052-588-6298
■札幌地区 TEL. 011-757-8799 FAX. 011-757-8999

企業・法人向けオリジナルPC情報はココで!
http://houjin.tsukumo.co.jp/original/

取扱店舗

東京地区

ツクモパソコン本店	03-3253-5599
ツクモDOS/Vパソコン館	03-3254-3999
ツクモ神田西口店	03-5207-9900
ツクモ高田馬場店	03-5292-0799
ツクモ八重洲店	03-3548-0399
ツクモ下北沢店	03-5465-0899

大阪地区

ツクモなんばParts王国	06-4396-6899

名古屋地区

ツクモ名古屋1号店	052-263-1655
ツクモ名古屋2号店	052-251-3399
ツクモ名駅店	052-561-0599
ツクモ栄店	052-238-7399

札幌地区

DEPOツクモ札幌駅前店	011-757-3299

TSUKUMO
www.tsukumo.co.jp

インテル®Pentium®4プロセッサ搭載「TS-JADE」

Intel® Pentium® 4プロセッサ 1.90GHz搭載

●インテル®850チップセット採用
●高い放熱性、メンテナンスが容易なオリジナル・アルミケース採用
●最新高速ビデオカード搭載

拡張性重視 上級モデル

●オリジナルアルミケース(AL-999/330USB)

TS-JADE 1900

JustLink/JustSpeed搭載 COMBO DRIVEモデル

●CPU:インテル®Pentium®4プロセッサ 1.90GHz
●マザーボード:インテル®850チップセット搭載
●メインメモリ:256MB(128MB×2/PC800RIMM/最大:2GB)
●ビデオカード:nVIDIA®GeForce3™ Ti200/64MB (DDR SDRAM)
●ハードディスク:60GB(UltraATA100/7200rpm)
●COMBO DRIVE:書込20倍速、書換10倍速、読込40倍速、DVD読込12倍速 B's Recorder GOLD & B's CLIP & Win DVD添付
●フロッピー:2モード
●ブロードバンド:10BASE-T/100BASE-TX
●サウンド:AC'97準拠(オンボード)
●キーボード:日本語109キー(PS/2)
●マウス:スクロールマウス(PS/2)
●ケース:W223×D493×H473mm(AL-999/330USB)
●電源:330W

Windows XP Homeモデル
TS-1900/60DCXP ………… **¥165,999**

※Intel、Intel Insideロゴ、Pentiumはアメリカ合衆国およびその他の国におけるインテルコーポレーションまたはその子会社の商標または登録商標です。

最新のIntel® Pentium® 4プロセッサ 2.00GHz搭載モデル登場!

ハイパフォーマンス
ハイスピード
ハイスペック

●余裕のパワーで多目的に対応

●スチールケース(H600A)

TS-JADE 2000

JustLink/JustSpeed搭載 CD-R/RW DRIVEモデル

●CPU:インテル®Pentium®4プロセッサ 2.00GHz
●マザーボード:インテル®845チップセット搭載
●メインメモリ:512MB(PC133 CL3/最大:3GB)
●ビデオカード:nVIDIA®GeForce2 MX™400/64MB(SDRAM)
●ハードディスク:80GB(UltraATA100/7200rpm)
●CD-R/RW DRIVE:書込20倍速、書換10倍速、読込40倍速、B's Recorder GOLD & B's CLIP添付
●フロッピー:2モード
●ブロードバンド:10BASE-T/100BASE-TX
●サウンド:AC'97準拠(オンボード)
●キーボード:日本語109キー(PS/2)
●マウス:PS/スクロールマウス(PS/2)
●ケース:W203×D470×H460mm(H-600A)
●電源:300W

Windows XP Homeモデル
TS-2000/80ARXP ………… **¥159,999**

Windows 2000 Professionalモデル
TS-2000/80ARS ………… **¥169,999**

TS-JADE 1900

JustLink/JustSpeed搭載 CD-R/RW DRIVEモデル

●CPU:インテル®Pentium®4プロセッサ 1.90GHz
●マザーボード:インテル®845チップセット搭載
●メインメモリ:256MB(PC133 CL3/最大:3GB)
●ビデオカード:nVIDIA®GeForce2 MX™400/64MB(SDRAM)
●ハードディスク:60GB(UltraATA100/5400rpm)
●CD-R/RW DRIVE:書込20倍速、書換10倍速、読込40倍速、B's Recorder GOLD & B's CLIP添付
●フロッピー:2モード
●ブロードバンド:10BASE-T/100BASE-TX
●サウンド:AC'97準拠(オンボード)
●キーボード:日本語109キー(PS/2)
●マウス:PS/スクロールマウス(PS/2)
●ケース:W210×D470×H400mm(AL-588/330USB)
●電源:330W

Windows XP Homeモデル
TS-1900/60ARXP ………… **¥129,999**

Windows 2000 Professionalモデル
TS-1900/60ARS ………… **¥139,999**

ご注文専用フリーダイヤル 受付時間/10:45～19:00 定休日/なし
☎ 0120-377-999

FAX24時間受付
FAX 03-3255-4199

ご注文はインターネットでも受け付けております。
http://shop.tsukumo.co.jp/original/

いよいよ目前！

20点確保!!

大学入試センター試験対策直前マスター

☆☆☆ 集中連載 その2 ☆☆☆

数学①②「BASIC言語」の予想問題と解法

東幸太

先月に引き続き、頻出問題を練習！

1月19日（土），20日（日）は，平成14年度大学入試センター試験の実施日。受験生のみなさんは，受験票も届き，いよいよ目前という感じだろう。この時期は新しい項目に手を出すよりも，すでに学習してあるポイントをしっかりと自分のものにする復習が有効である。自分の守備範囲を100%固めておき，設問がその範囲に入った場合は確実に満点を取れるようにしておきたい。

「数学①（数学Ⅰ・数学A）」「数学②（数学Ⅱ・数学B）」の選択問題にはBASIC言語を使った問題が出題される。ここでは，先月号に引き続き，基本ポイントを復習したうえで，問題の頻出パターンをいくつか練習しておこう（先月号を読んでいない人は，今すぐバックナンバーを注文しよう）。

BASIC 言語問題おさえておくべきポイント

(1) FOR - NEXT のループ
(2) IF - THEN の条件判断
(3) INT()関数の動作とINT関数利用による剰余計算・約数判断
(4) サブ・ルーチンの動作
(5) 配列変数の利用
(6) フローチャートとプログラムの対応関係

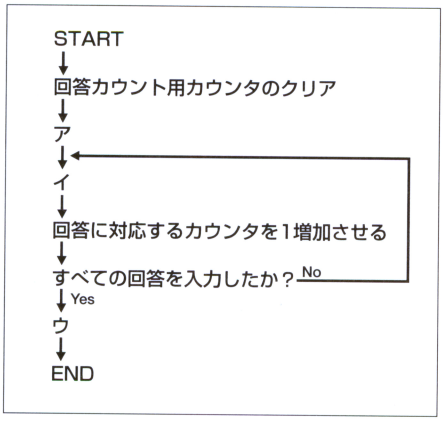
（※予想問題1，2は1月号で掲載しています）

予想問題 3

アンケートの回答を集計することを考える。アンケートには一つの質問があり，五つの候補の中から一つを選択するようになっており，回答は1から5までの整数となる。

アンケート回答数を最初に入力し，その後すべてのアンケート回答を入力して，1から5までの回答候補それぞれについて回答数がいくつあったかを集計，表示したい。

集計プログラムのフローチャートとして以下のようなもの（第1図）を考えた。これに関して下記の問いに答えよ。



《第1図》
設問のフローチャート

[1] アからウまでに相応しいものを以下から選び，フローチャートを完成させよ。

(1) 回答入力
(2) 回答に対応しないカウンタのクリア
(3) 集計結果表示
(4) 回答表示
(5) 回答総数入力

[2] フローチャートをもとにして以下のようなプログラムを作成した。

```
10 DIM A(5)
20 FOR B=1 TO 5
30  A(B)=0
40 NEXT
```

```
50 INPUT "N=";N
60 FOR B=1 TO N
70  INPUT "Ans.=";D
80  A(D)=A(D)+B
90 NEXT
100 FOR B=1 TO 5
110  PRINT "Ans.";B;" =";A(B)
120 NEXT
130 END
```

このプログラムを実行し，「N=?」の表示に対して5を入力した。その後，5回表示された「Ans.=?」に対して，順に「5，2，2，2，3」と入力した。最終的に表示される結果は，

Ans.1 = エ
Ans.2 = オ
Ans.3 = カ
Ans.4 = キ
Ans.5 = ク

となる。

[3] 正しい結果が得られるようにプログラムを以下のように修正した。ケからサまでに相応しいものを以下から選び，プログラムを完成させよ。

```
10 DIM A(5)
20 FOR B=1 TO 5
30  A(B)=0
40 NEXT
50 INPUT "N=";N
60 FOR B=1 TO ケ
70  INPUT "Ans.=";D
80  A(D)=A(D)+コ
90 NEXT
100 FOR B=1 TO 5
110  PRINT "Ans.";B;" =";A(サ)
120 NEXT
130 END
```

ケ，コ，サについての選択肢
（1）0 （2）1 （3）5 （4）A （5）B （6）D （7）N

解法3

過去の出題パターンを分析すると

パソコンがなければ実際に動作させることができないBASICプログラミングについて，筆記試験を行うというのは基本的にやや無理がある。このため，センターテストに出題されるBASIC問題には変化を与えることがむずかしく，ここ数年の問題にはフローチャートを組み合わせたものや，プログラムの修正を前提としたものなども登場するようになった。

問題のパターンが一般的なものとは異なるため，見かけはやっかいに感じられるかもしれないが，BASICプログラミングについての理解ができていれば，さほど問題にはならないはずである。新しいパターンの問題で驚かされることがないようにしたい。

フローチャートとは BASICプログラムの表現を変えたもの

この問題では，最初の設問がフローチャートに関するものになっている。フローチャートはプログラムで処理すべき内容を人間の言葉で記述したものであり，処理内容自体はプログラムと1対1で対応する。よって，一般的なBASICプログラムの問題の表現を変えたものと考えれば良い。よって，フローチャートの抜けている部分を埋める穴埋め問題は，プログラムの抜けている部分を埋める穴埋め問題と同じということになる。

フローチャートを見ると，

- 回答カウント用カウンタクリア
- 回答に対応するカウンタのカウントアップ
- すべての回答を入力したかどうか

という三つの要素が含まれているが，これだけでは設問の望んでいるプログラムは要素が不足していて作成できない。

ここで，選択肢を見てみると，五つの要素が用意されている。よって，これらの中の三つを適切な空欄に当てはめ，アンケート集計プログラムのフローチャートを完成させれば良い。問題の文章を見ると，「アンケート回答数とアンケート回答を入力」や，「回答数がいくつあったかを集計」，「表示したい」といった部分があり，これらはプログラムに含まれていなければならず，当然フローチャートにも含まれている必要がある。

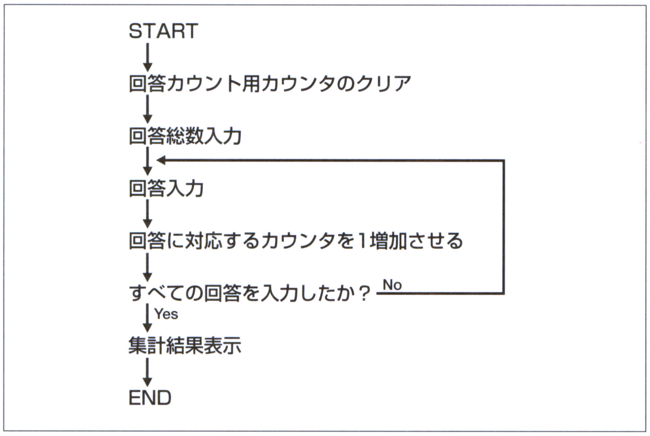
ループと抜け出た処理の部分は 何を行っているのか考える

フローチャートの構成を見ると，一つだけループ構成が含まれている。ループ内部にはイとカウントアップの処理が入っており，すべての回答を入力するまで，この二つの処理を実行する形になっている。すべての回答を入力したかどうかは，回答

数がわからない限り判断できないことを考えると，ループに入る前の段階で回答数が用意されている可能性が高い。ループの前段では，回答カウント用カウンタクリアとアの処理が実行されており，アの部分で回答数が決まると予想できる。ループ処理の部分が集計のループになっていることから，ループ内部ではアンケートの回答を入力していると考えられる。また，ループから出た段階での処理はウの処理だけであるが，ループから出た後の処理には結果の表示要素が含まれるはずである。これらの予想から相応しい選択肢を当てはめていくと，

ア（5）
イ（1）
ウ（3）

となる。これ以外の選択肢の（2）と（4）を見ると，プログラムに必要とされている要素に該当していないことがわかり，上記の回答が正しいことが確認できる。解答をあてはめたフローチャートは第2図のようになる。



《第２図》解答をあてはめたフローチャート

設問から推定した結果が利用できないバグを含んだプログラム

［2］の設問では，フローチャートをもとにしてプログラムを作成している。このプログラムを実行し，アンケート総数として5を入力し，回答として5，2，2，2，3の五つの数値を入力した結果が問われている。

プログラムの動作はあらかじめわかっているため，集計結果として

エ　0
オ　3
カ　1
キ　0
ク　1

となることがわかる。ところが，設問［3］を見ると，このプログラムにはバグがあり，想定した動きをしていないことが判明する。このため，プログラムの動作をもとにした推定結果は利用できず，実際にこのプログラムをシミュレートしていく必要がある。

順番にプログラムを実行してみる

10行では配列変数Aを宣言しており，20行から40行の部分で添え字1から5に対応する変数を0クリアしている。50行では入力処理が行われており，アンケート総数を入力していると考えられる。つぎの60行で，FOR-NEXTループの範囲が1からNまでになっていることから，この予想が裏付けられる。

70行ではアンケート回答を入力しており，80行でアンケート回答数値に対応する配列変数Aの値をループ変数Bだけ増やしている。アンケート回答入力が終了すると100行から120行で配列変数Aの添え字1から5に対応する値を表示している。

各変数を地道に調べていく

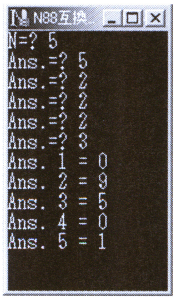
最終的に表示されるのは配列変数Aの値であることから，Aの値の変化に注目する。Aを変化させているのは80行だけであるため，80行を実行する段階での各変数の値を調べていく。

ここではループ変数Bが加算されているため，一つ目の回答の「5」を入力している段階ではA（5）は1増加する。ところが，二つ目の回答の「2」を入力している段階ではループ変数Bが2になっているため，A（2）は2増加する。三つ目ではA（2）が3増加し，四つ目では4それぞれ増加する。最後にA（3）が5増加してループが終了する。

80行実行回数	B	D	A(2)	A(3)	A(5)
1回目	1	5	0	0	1
2回目	2	2	2	0	1
3回目	3	2	5	0	1
4回目	4	2	9	0	1
5回目	5	3	9	5	1

結果として，

エ　0
オ　9
カ　5
キ　0
ク　1



◀設問2の実行画面（あきらかにまちがいがわかる）

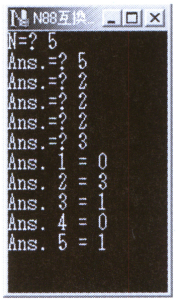
が解答となる。フローチャートとプログラムの対応が取れておらず，問題やフローチャートなどから期待される結果になっていないのである。

プログラムをシミュレートすれば簡単だが，動作確認はしっかり

［3］の設問では，プログラムを修正し，フローチャートと正

しく対応させている。プログラムにはケ，コ，サの三つの空欄があり，その部分に正しいプログラムを当てはめる必要がある。設問[2]でのシミュレーション結果から，80行での処理が不適切で，A（D）には1を加えるだけで良いことがわかる。他の部分はそのままで良いため，

ケ（7）N
コ（2）1
サ（5）B



◀プログラムを修正して実行した画面

が解答となる。

解答ができたら，実際にシミュレーションを行い，問題やフローチャートと対応した動作をするかどうかを確認する。40行までで集計用の配列変数を初期化し，50行でアンケート総数を入力，60行から90行で，アンケートの結果を入力し，対応する配列変数をカウントアップさせていく。最後に100行から120行で結果を表示する。この結果から，プログラムが正しく動作することがわかり，解答が正しいことが裏付けられる。

デバッグ作業を応用した問題は昨年も出題された！

ミスを含むプログラムを修正させるデバッグタイプの問題は，昨年のセンター試験に出題されている。実際のプログラミングでは頻繁に行われるデバッグを，試験問題に応用しているのだが，デバッグというプロセスになじみがないと，ややむずかしく感じられるかもしれない。しかし，試験問題のような小規模プログラムであれば，自分なりにプログラムを作って，ちがいを見つけるといった手法でも十分対応可能である。

本質的部分の難易度は高くならないと思われるので，あわてずに対応したい。

予想問題4

下記のプログラムは，入力された2以上50以下の自然数AとBを素因数分解し，より大きな素因数を持っているほうの数値(AまたはB)を表示するものである。このプログラムに関する以下の問いに答えよ。

```
10 INPUT "A=";A
20 INPUT "B=";B
30 X=A
40 GOSUB アイウ
50 C=Y
60 X=B
70 GOSUB エオカ
80 D=Y
90 IF C>D THEN PRINT A
100 IF C<D THEN PRINT B
110 IF C=D THEN PRINT "Same"
120 END
130 FOR I=2 TO 50
140  IF INT(X/I)*I<>X THEN GOTO キクケ
150  X=X/I
160  Y=I
170  GOTO コサシ
180 NEXT
190 RETURN
```

[1] プログラム中の空欄に相応しい行番号をあてはめてプログラムを完成させせよ。

[2] Aとして6，Bとして4を入力した場合，140行のIF文はスセソ回実行され，そのうちタチツ回は条件が成立せずに150行から170行が実行される。また，190行のRETURN文はテトナ回実行される。

[3] Aとして35，Bとして40を入力した際に，160行の代入文はニヌネ回実行される。

解法4

実際のプログラミングで多用されるGOSUB-RETURN文もチェック！

BASIC問題に出題される可能性がある要素として，GOSUB-RETURN文を利用したサブ･ルーチンが挙げられる。サブ･ルーチンは似た処理を一つにまとめ，プログラムを短くして見やすくする効果がある。実際のプログラミングでは「関数」という形で多用されている技術となっている。

ジャンプ先の行番号を求める問題は流れをじっくり読み取る

最初の設問は，プログラム中の空欄を埋める問題である。空欄はすべてGOSUB文やGOTO文のジャンプ先行番号となっている。よって，本来であれば問題の条件をもとにプログラムの流れをすべて把握することで解答を導き出すことになる。しかし，今回の問題の場合，BASICの知識さえあれば，一部の空欄について正解を求めることが可能である。

まず，GOSUB文のジャンプ先のアイウとエオカであるが，候補としてはRETURN文で復帰することができる130行から190行が考えられる。この中で，FORループ内部に該当する140行から180行にジャンプしたとすると，FOR文を実行する前にNEXT文を実行してしまいエラーとなる。140行と170行のジャンプ先によっては，エラーにならない可能性も残ってはいるが，非常に不自然なプログラムとなってしまう。

また，RETURN文だけの190行をGOSUB文で呼び出すことも無意味である。この結果，残る候補は130行だけになる。あくまでもこの段階では候補であり，正式な解答にするかどうかは最後に動作確認することで判断する。

無限ループになるなど，あり得ない行を消していく

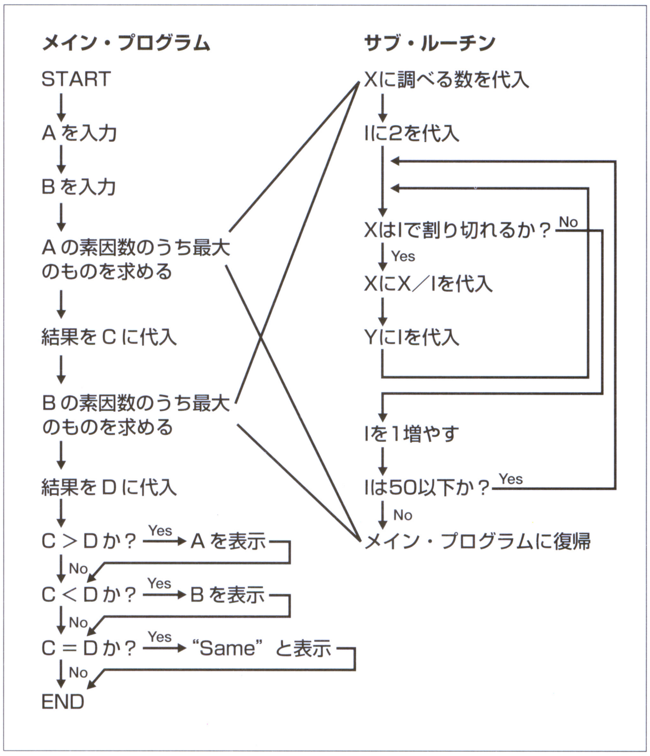
つぎの空欄は140行のIF文条件成立時のジャンプ先キクケであるが，それよりも簡単な170行のジャンプ先のコサシに着目する。コサシのジャンプ先候補としては，130行から190行があるが，150行から170行は無限ループになってしまうのであり得ない。また，130行はFOR文を複数回実行することになるので不自然である。180行にジャンプするのは無意味である。よって，残る候補は140行と190行になる。ループを継続するか，ループから抜け出してサブ・ルーチンから復帰するか，のどちらかになる。

ここでキクケについて考えてみると，コサシが140行であれば，キクケは140行から170行に構成されるループを抜け出すジャンプということになり，180行で確定する。これは，190行ではFOR文を使う意味がなくなるためである。また，コサシが190行であれば，キクケはループを継続するためのジャンプということになり，ほぼ180行で確定する。つまり，キクケはどのみち180行になるわけである。

このように，十分なBASICプログラムの知識があると，問題に含まれる数学的要素を無視したまま，ある程度解答を求めることができる。

数学的知識も若干要求される場合もある

コサシのジャンプ先については，問題の数学的要素を検討する必要がある。90行から110行の部分で最終的な表示を行う際には変数CとDが参照されているが，これらの変数にはサブ・ルーチン実行後の変数Yの値が代入されており，その変数Yはサブ・ルーチンの160行で代入されている。表示に関する条件は「より大きな素因数を持っている」かどうかであり，変数CやD，Yが，より大きな素因数を返してきていることが予想できる。140行では変数XがIで割り切れるかどうかを判断しているが，このIは2から50までカウントアップされていく。よって，より大きな素因数を調べるためには，割り切れて150行以降が実行されても，ループを継続してより大きな素因数がないかどうかを確認しなくてはならない。この点から，コサシはループを継続するための140行となる。フローチャートは第3図のようになる。



《第３図》予想問題４のフローチャート

穴埋め問題は複数の設問を同時にシミュレートしたほうがよい

プログラムの空欄を埋めるプログラミング系の問題の場合，プログラムができあがったら問題で示されている動きをするかどうかをシミュレートして確認する必要がある。正しく動作する場合は，解答はすべて正しいと見てよい。これは，規模の小さいプログラムの場合は1か所でもミスがあれば，動作が明確におかしくなるためである。

実際に適当な数値をAとBに設定してシミュレートを行ってもよいが，つぎの設問を見るとA＝6，B＝4のときの動作をシミュレートすることになりそうである。よって，［2］の設問を解くためのシミュレートと，設問［1］の解答確認のためのシミュレートを同時に行うことにする。

設問［2］では，140行のIF文の動作が問われている。よって，

140行に着目してシミュレートを進めていく。10行から120行のメイン・プログラムは基本的に一本道であるため，40行と70行で呼び出される130行以降のサブ・ルーチン部分を調べていく。

40行からの1回目のサブ・ルーチン呼び出し

X	I	Y	140行の動作
6	2	初期値	6は2で割り切れるので不成立　X＝3，Y＝2となる
3	2	2	3は2で割り切れないので成立
3	3	2	3は3で割り切れるので不成立　X＝1，Y＝3となる
1	3	3	Xが1なので2以上のIに対しては条件は常時成立する
			I＝4から50までループが継続される
		3	最終的にY＝3となってメイン・プログラムに復帰

70行からの2回目のサブ・ルーチン呼び出し

X	I	Y	140行の動作
4	2	3	4は2で割り切れるので不成立　X＝2，Y＝2となる
2	2	2	2は2で割り切れるので不成立　X＝1，Y＝2となる
1	2	2	Xが1なので2以上のIに対しては条件は常時成立する
			I＝3から50までループが継続される
		2	最終的にY＝2となってメイン・プログラムに復帰

1回目の呼び出しでY＝3，2回目の呼び出しでY＝2となることから，Cは3，Dは2となり，90行から110行のIF文は90行のみ成立する。結果としてAの値の6が表示される。6は2×3，4は2×2であることから，「より大きな素因数を持っている」数値は6であり，正しい動作をしていることがわかる。

この結果から，設問［1］での解答の裏付けが取れる。

アイウ＝130　**キクケ＝180**
エオカ＝130　**コサシ＝140**

プログラムを確実にシミュレートしていけば大丈夫

さて，設問［2］について見ていくと，140行のIF文は140行が常に成立しても1回のサブ・ルーチン呼び出しに対して49回実行される。それに加えて，IF文が不成立だった場合には不成立になる回数だけ実行される回数が増加する。これは，不成立時にはIがカウントアップされないためである。結果的に，1回目の呼び出しで2回，2回目の呼び出しで2回，それぞれ条件が不成立になるため，IF文の実行回数は

49＋2＋49＋2＝102

となり，

スセソは102となる。

また，条件が成立しないのはそれぞれ2回であり，合計で4回となる。よって，

タチツは004となる。

190行のRETURN文は，サブ・ルーチンからの復帰のときに実行される。サブ・ルーチンはメイン・プログラムから2回しか呼び出されないため，実行回数は2回であり，

テトナは002となる。

設問［3］では，A＝35，B＝40の場合の160行の実行回数が問われている。ループ部分の処理が比較的簡単なため，単純にプログラムをシミュレートすることも不可能ではない。しかし，より時間を節約するためにはプログラムの動作から実行回数を考えるほうが有利である。

設問［2］でのシミュレーション結果を考えてみると，160行が実行されるのは140行のIF文が不成立になった場合である。不成立になるというのは，XがIで割り切れたときであり，その場合は150行でXをIで割った後，Iを変化させずに再度140行を実行することになる。つまり，Xを約数で割ったときに150行から170行が実行されるのである。Iは2から順番に，割り切れないときにカウントアップされていくため，除数は常に素因数となる。よって，サブ・ルーチンを呼び出す前にXに設定した数値について，その数値を素因数分解した結果の素因数の個数回だけ実行されることになる。35と40を素因数分解すると，

35＝5×7

40＝2×2×2×5

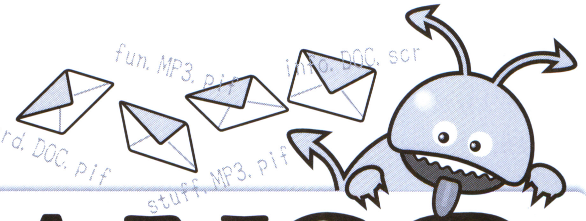
となり，合計六つの素因数が含まれることから160行は6回実行されることになる。よって，

ニヌネは006となる。

最後に

配列変数とサブ・ルーチンは，実際のパソコンがない状態で解答しなければならないセンター試験には出しづらい要素であるが，プログラミングの面から見るとループや条件判断と並ぶ極めて重要な要素である。出題されたときにあわてないためにも，その動作について確認しておきたい。

センター試験の現場では，これまでになく集中力が重要となる。同時に，どこに集中させればよいかを判断する観察力も必要である。最初はマクロな視点から問題の雰囲気を調べつつ意識を整え，その後ミクロな個々の設問に全力で集中していきたい。健闘を祈る。



感染してもあわてない!!
コンピュータ・ウイルス「BADTRANS.B」撃退法

コンピュータ・ウイルス「BADTRANS.B」の被害が急増中!! ある日突然「ウイルスに感染してますよ!!」というメールが届いたら，あなたは対処できますか？　実際にBADTRANS.Bの駆除方法を紹介しますので，非常時の参考にしてください。

文／斉藤大地
協力／トレンドマイクロ
シマンテック

「BADTRANS.Bに感染して困っている人は今すぐ18ページへ」

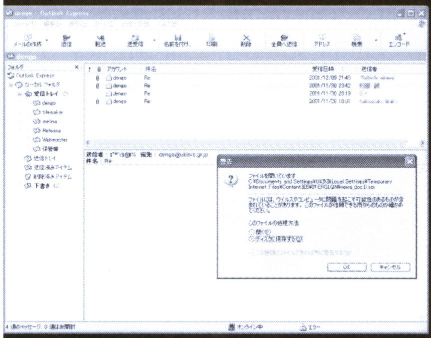
忍び寄るウイルス

先月号で紹介したNimda（ニムダ）に引き続き，BADTRANS.B(バッドトランス・ビー)と呼ばれるコンピュータ・ウイルスが大流行しています。感染したユーザーは“どうして感染したんだろう…”と不思議に思う人も多いようですが，今回もまたInternet Explorer（IE）のセキュリティホールが利用されたせいで気付かずに感染してしまったのです。

今回のBADTRANS.Bで使われた手口がNimdaと似ていることからもわかるように，コンピュータ・ウイルスを作る犯人もウイルスをどうやって流行させるか対策を考えているようです。

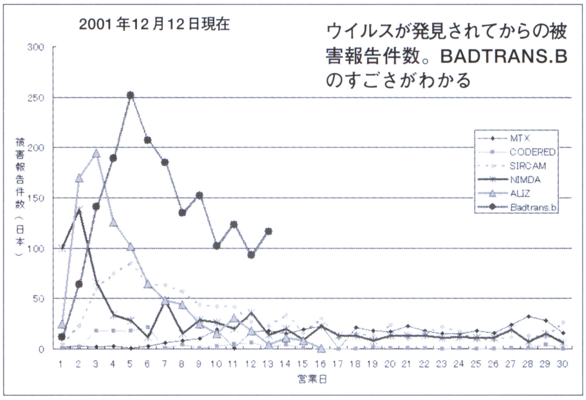
BADTRANS.Bは，正式には「WORM_BADTRANS.B」(ワーム・バッドトランス・ビー）などと呼ばれる迷惑プログラムの一種。ワームは厳密にいうと，ほかのプログラムやデータ・ファイルに寄生するコンピュータ・ウイルスとはちがうのですが，社会的影響はかなりのもの。そのために，現在ではウイルス同様にあつかわれています。

相手に届くHTMLメール



▲BADTRANS.Bが送るメールは，このようなダウンロードのウインドウが勝手に開く

ウイルス被害件数グラフ（トレンドマイクロ提供）



BADTRANS.Bでどうなる？

このBADTRANS.Bに感染して発病（プログラムが起動）すると，下記のような症状があらわれます。

【自己増殖（ワーム）活動】

BADTRANS.Bは増殖活動として，電子メールを使ってプログラムのコピーを送信する活動を行います。このとき，Windows向け電子メールソフトのほとんどが利用する「MAPI」と呼ばれるインターフェースを利用して，Windowsに設定されている電子メール情報を参照。ほかの電子メールソフトに頼らずにウイルスが添付されたメールを自身で送信しますから，電子メールソフトの履歴など(Outlook Expressでは送信済みアイテム）には痕跡が残りません。

ウイルスによって送られる電子メールは，Internet Explorerのセキュリティホールを利用するHTMLメール。対策がなされてないと，プレビューしただけで添付されたウイルス・プログラムが起動してしまいます。言うまでもありませんが，対策を施していてもプログラムを実行すれば感染する可能性があります。

【ハッキング活動】

プログラムが起動してメモリに常駐すると，キー入力の内容を時間とともに記録します。これらの内容は，あるメール・アドレスに送信されるようになっていますから，IDやパスワードなどの重要な情報が外部に漏れる可能性があります。

知らない人から感染の事実を知らされるのはなぜ？

BADTRANS.Bがメール送信に利用するのは，「受信トレイ」内のメールだけではありません。「マイドキュメント」と「temporary Internet files」フォルダ内にある，拡張子が“ht”ではじまる文書と“asp”の文書内にあるメール・アドレスも利用されてしまいます。

つまり，あなたがホームページを見ただけでもキャッシュとしてディスク内に残ったファイルから，電子メール・アドレスが参照されてしまうのです。このため，見ず知らずの人に迷惑をかけてしまい，感染の事実を知らされることになるのです。

BADTRANS.Bを駆除しよう!!

ウイルス対策ソフトを販売する各社では，急激な広がりを見せるBADTRANS.Bの社会的な影響を考慮して，修復・駆除ツールの無償配布を開始しています。今回は，トレンドマイクロ社が無償配布している「WORM_BADTRANS.B　修復ツール」を利用して実際にウイルスの駆除を行ってみます。

なお，このプログラムはその名のとおり「WORM_BADTRANS.B」というウイルスに対してのみ有効です。そのほかのウイルスやBADTRANS.Bに亜種が出現した場合，有効でない可能性がありますのでご注意ください。

症状の再確認を

BADTRANS.Bに感染したために現れる症状で，ユーザーに見分けがつきやすいのが「ウイルス・メール」です。BADTRANS.Bによるウイルス･メールにはつぎのような特徴があります。

- タイトルが：「info」「docs」「Humor」「fun」のいずれか，あるいは「Re:」となっている
- 本文が空白で中身がない
- 二つ拡張子がついた添付ファイルがある
- メール送信者の正規アドレスに「_」が追加されている（例：_webmaster@basicmagazine.net）

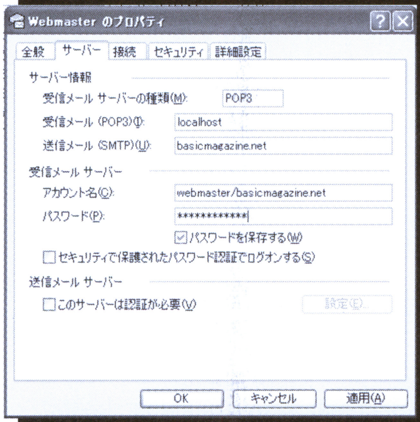
このようなメールが届いたと報告された場合，パソコンはBADTRANS.Bに感染している可能性が濃厚です。

駆除ツールのダウンロード

駆除ツールをダウンロードする前に注意したいのは，感染したパソコンでインターネットへ接続すると，その間にもウイルス・メールが送信されること。できれば，感染していないパソコンで駆除ツールをダウンロードし，フロッピーディスクなどで感染したパソコンに駆除ツールを持ってきたいものです。

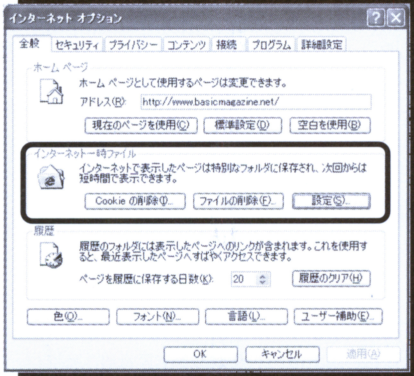
もし，ほかにパソコンがない場合は，インターネットへ接続する前に普段利用する電子メールソフトからメールサーバーやアドレスなどの設定をいったん消去し，あわせて，キャッシュを消去して被害の拡大を防ぎます（設定は忘れないようにメモを取るなどしましょう）。

メーラーの設定



▲普段使う電子メールソフトがOutlook Expressなら「ツール」メニュー内にある「アカウント」にメールサーバなどの情報がある

キャッシュの消去



▲キャッシュの中からメール･アドレスを参照されないように，ファイルを消去しておこう

届いたメールのソース

```
RETURN-PATH: <WEBMASTER@BASICMAGAZIN.NET>
DELIVERED-TO: KAGE@BASICMAGAZINE.NET
RECEIVED: (QMAIL 14321 INVOKED FROM NETWORK); 9 DEC 2001 21:48:54 +0900
RECEIVED: FROM UNKNOWN (HELO T-MTA4.BASICMAGAZINE.NET) (192.168.0.1)
  BY UKIPRO.GR.JP WITH SMTP; 9 DEC 2001 21:48:54 +0900
RECEIVED: FROM AOL.COM ([192.168.0.5]) BY T-MTA4.BASICMAGAZINE.NET WITH SMTP
          ID <20011209124840902.EBGY.3596.T-MTA4.BASICMAGAZINE.NET@MTA4.BASICMAGAZINE.NET>
          FOR <KAGE@BASICMAGAZINE.NET>; SUN, 9 DEC 2001 21:48:40 +0900
FROM: "WEBMASTER" <_WEBMASTER@BASICMAGAZINE.NET>
TO: KAGE@BASICMAGAZINE.NET
SUBJECT: RE:
MIME-VERSION: 1.0
CONTENT-TYPE: MULTIPART/RELATED;
              TYPE="MULTIPART/ALTERNATIVE";
      BOUNDARY="====_ABC1234567890DEF_===="
X-PRIORITY: 3
X-MSMAIL-PRIORITY: NORMAL
X-UNSENT: 1
MESSAGE-ID: <20011209124840902.EBGY.3596.T-MTA4.BASICMAGAZINE.NET@MTA4.BASICMAGAZINE.NET>
DATE: SUN, 9 DEC 2001 21:48:53 +0900
STATUS:  U
X-UIDL: 1007902134.14327.ST73.ARENA.NE.JP

--====_ABC1234567890DEF_====
CONTENT-TYPE: MULTIPART/ALTERNATIVE;
      BOUNDARY="====_ABC0987654321DEF_===="

--====_ABC0987654321DEF_====
CONTENT-TYPE: TEXT/HTML;
                CHARSET="ISO-8859-1"
CONTENT-TRANSFER-ENCODING: QUOTED-PRINTABLE


<HTML><HEAD></HEAD><BODY BGCOLOR=3D#FFFFFF>
<IFRAME SRC=3DCID:EA4DMGBP9P HEIGHT=3D0 WIDTH=3D0>
</IFRAME></BODY></HTML>
--====_ABC0987654321DEF_====--

--====_ABC1234567890DEF_====
CONTENT-TYPE: AUDIO/X-WAV;
                NAME="SETUP.DOC.SCR"
CONTENT-TRANSFER-ENCODING: BASE64
CONTENT-ID: <EA4DMGBP9P>

TVSMAAKEAAAEAAAA//8AALGAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

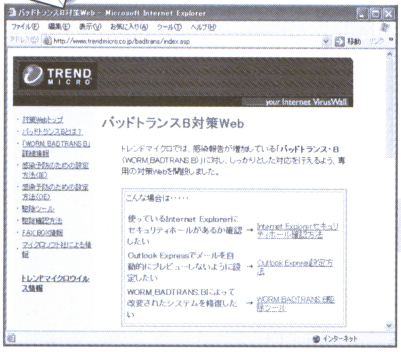
← インラインフレームのソースとしてウイルスが読み込まれる（<IFRAME SRC=3DCID:EA4DMGBP9P HEIGHT=3D0 WIDTH=3D0>）

← WAVEのデータ・ファイルだと偽わっている（CONTENT-TYPE: AUDIO/X-WAV;）

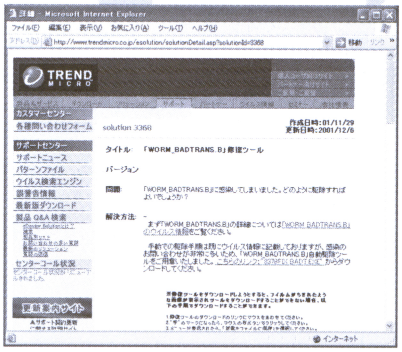
← このファイルが開かれる（NAME="SETUP.DOC.SCR"）

▲見た目では何もない真っ白なメール。でも，裏ではプログラムが起動するしかけが…（注：筆者へ実際に届いたメールを加工しています）

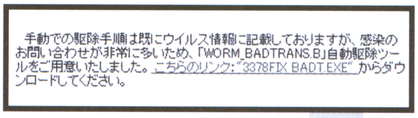
駆除ツールのダウロード開始



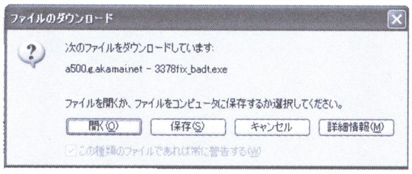
トレンドマイクロ社の「バッドトランスB対策Web（http://www.trendmicro.co.jp/badtrans/）」にアクセス



左のメニューから「駆除ツール」を選択すると新しくウインドウが表示される



文章の中ほどにある「こちらのリンク…」と書かれたところをクリック

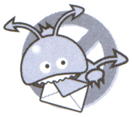


駆除ツールのダウンロードがはじまるので「保存」を選択

駆除ツールのログ（FIX_BADT）

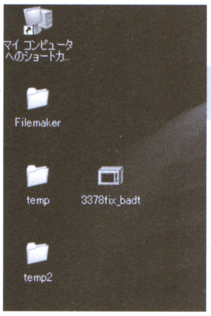
```
================================================================
START : FIX_BADTRANS VERSION 1.10
  WINDOWS PLATFORM : WINDOWS 98
  TOOL CONFIGURATION:
            SCAN FILES:                    YES
            QUIET MODE:                    YES
DATE AND TIME: MON DEC 10 08:22:43 2001

I. TERMINATING WORM/VIRUS PROCESSES.
SCANNING [C:\WINDOWS\SYSTEM\KERNEL32.EXE] -> FOUND WORM_BADTRANS.B.  TERMINATED.
   *** INFECTED PROCESSES WERE DETECTED. ***

II. CLEANING SYSTEM REGISTRY.
 WORM_BADTRANS.B FOUND IN THE "SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUNONCE\KERNEL3
2"---> REMOVED

III. REMOVING DROP FILES.
 FILE "C:\WINDOWS\SYSTEM\KERNEL32.EXE" DELETED
   FILE "C:\WINDOWS\SYSTEM\KDLL.DLL" DELETED
 FILE "C:\WINDOWS\SYSTEM\CP_25389.NLS" DELETED

IV. INOCULATING THE SYSTEM.
  CREATING [KERNEL32.EXE].  FOLDER  CREATED.
                              SYSTEM INOCULATED

STOP
DATE AND TIME: MON DEC 10 08:22:48 2001
================================================================
```

▲作成された「FIX_BADT」の中身がこれ。駆除ツールがどのような処置を行ったかわかる

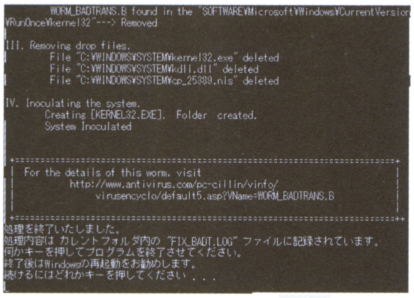


修復ツールの利用方法

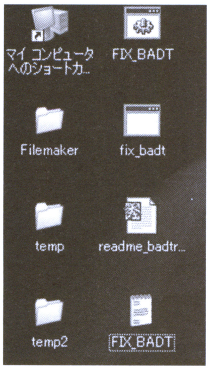
ダウンロードした駆除ツールの利用方法は非常に簡単です。



ダウンロードした駆除ツール「3378fix_badt」のアイコンをダブルクリック



ソフトが自動的に起動して駆除がはじまります



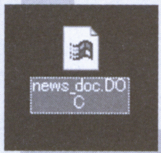
駆除が終了するといくつかファイルが作成されます。プログラムが終了したらシステムを再起動しましょう

以上で駆除は終了です。システムが再起動したらログ・ファイルとなる「FIX_BADT」の中身を確認しましょう。

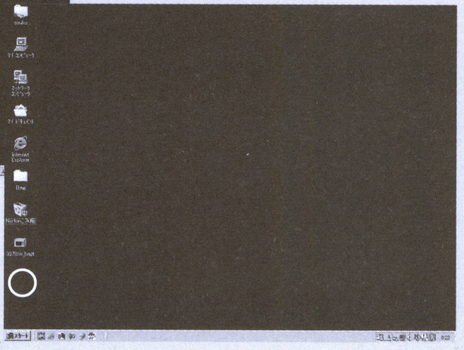
BADTRANS.Bを実行すると？

今回の記事執筆のために，あえてWindows98の環境を作り，ウイルス・プログラムを実行して試してみた。ウイルスが起動したせいで，特別エラー表示などが起こることはなかった。

つまり，セキュリティ対策をしていない限り，メールをプレビューしただけで気づかずにウイルス・プログラムが起動してしまうことになる



◀▲実際に送られてきたBADTRANS.Bをデスクトップに置いてみた。ファイル名は「news.doc.DOC」だ



▲ダブルクリックしたらデスクトップから消えてしまった

セキュリティの強化にチャレンジ!!

コンピュータ・ウイルスに限らず，ネットワーク経由でパソコンの情報を盗まれる被害にあうケースが多発し，セキュリティの低さが問題となっています。あなたのパソコンは大丈夫ですか？

一般ユーザーや学校が標的に

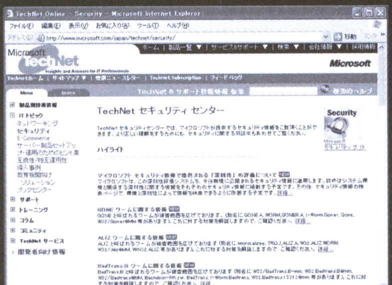
BADTRANS.Bで利用された手段は，以前にも流行した「Nimda（ニムダ）」ウイルスと同じ“不適切なMIMEヘッダが原因でInternet Explorer（IE）が電子メールの添付ファイルを実行する（MS01-020）”という一部バージョンのIEにあるバグを利用したもので，以前からその危険性が指摘されていたものでした。

企業などでは対策が施されていたところも多かったようですが，一般や学校のコンピュータでは管理が行き届かず，それが，BADTRANSの大流行を助長する原因になったようです。

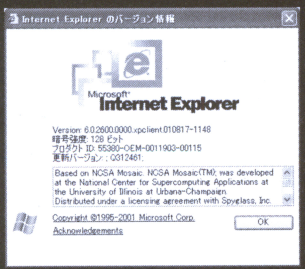
セキュリティを強化しよう

このバグを利用したコンピュータ・ウイルスは引き続き出現しそうですから，今回BADTRANS.Bを無事に駆除できても新たなウイルスに感染する可能性もあります。そこで，みなさんのパソコンの設定をもう一度確認してみましょう。

Microsoft TechNet セキュリティセンター（http://www.microsoft.com/japan/tech net/security/）



▲重大なバグやウイルスの情報が記載されるので，ときどきチェックしたい



IEのバージョン表示

◀バージョンが5.0/5.01/5.5ならアップデートが必要。ただし，下部に更新バージョンとして“SP2”の表示があれば対策済み

セキュリティ強化策

その1 バグを治す【必須!!】

この“不適切なMIMEヘッダが原因でInternet Explorerが電子メールの添付ファイルを実行する（MS01-020）”によって影響を受けるのは，IEのバージョン5.01と5.5。

いずれもService Pack 2（SP2と略される）では問題が改善されていますので，Windows Updateを利用してアップデートさせるか，マイクロソフト社のホームページ（http://www.microsoft.com/japan/technet/security/virus/badtrans.asp）を参考にしてIEをバージョン・アップさせましょう。

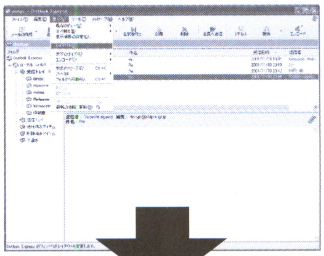
IEのバージョンを確認する方法は，IEを立ち上げた状態で「ヘルプ」メニューから「バージョン情報」を選択します。

この対策を行うと，ウイルスがHTMLメールでダイレクトにアクションを起こそうとしても「保存のダイアログ」が表示されるようになります。

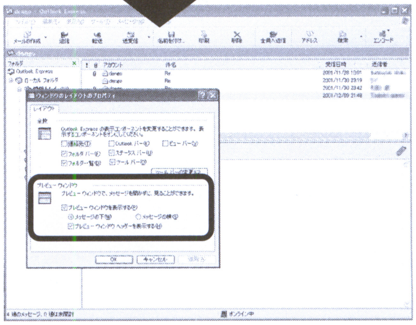
その2 メールのプレビューをやめる

BADTRANS.Bのようなウイルスは，IEをアップデートしてもプレビューしただけで保存のダイアログが表示されます。まったく気づかずにウイルスに感染することはなくなりますが，何かの拍子に誤って保存し，実行してしまうかもしれません。そこで，メールのプレビュー機能を停止して被害を防いでみましょう。

・プレビューを非表示に設定する方法



◀まず，「表示」メニューから「レイアウト」を選んだら…



▲下部にある「プレビューウィンドウを表示する」のチェックをはずします

以上で，設定は終了です。これなら，件名を選んでメッセージを表示させない限り「保存のダイアログ」が出ることはなくなります。

でも，これでは不便だと思う人もいるでしょう。それなら，ウイルス対策ソフトを導入しましょう。

その3 ウイルス対策ソフトの導入【推奨】

ウイルス対策ソフトを導入するメリットは，面倒なセキュリティ対策のほとんどを一手に引き受けてくれるところにあります。

人間に例えると，病気にかかる前に“うがい”や“予防薬”で病気自体を防ぐのがウイルス対策ソフト。一方，駆除ツールは病気で入院する羽目になり，注射や手術をしてもらうのに似ているでしょうか。

市販されているウイルス対策ソフトは，

《第1表》Internet Explore のバージョンとセキュリティホールの有無

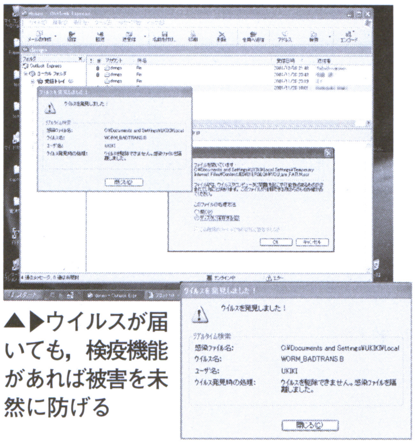
Internet Explorer バージョン	サービスパック	対応OS	同セキュリティホールの有無
5.0/5.01/5.5	なし	Windows※1	あり
5.01/5.5	Service Pack 1	Windows※1	あり
5.01/5.5	Service Pack 2	Windows※1	なし
6（最小構成）	——	Windows※1	あり
6（標準構成以上）	——	Windows※1	なし
——	——	Macintosh	なし
——	——	Windows XP	なし

※1――Windows XPを除く　　トレンドマイクロWebページより

コンピュータ・ウイルスに対する防御・駆除などにくわえ，最近では有害なホームページからあなたを守ってくれたり，ネットワーク経由で侵入してこようとするクラッカーなどからパソコンを守る機能も併せ持ちます。

多少の投資は必要ですが，安心してインターネットライフを送るためには必要不可欠な存在となりつつあるようです。

ウイルス対策ソフトのメール検疫



▲▶ウイルスが届いても，検疫機能があれば被害を未然に防げる

HTMLメールのデメリット

最近，多くのユーザーに利用されるようになったHTMLメール。カラフルで見やすい電子メールを送れる一方で，ウイルス問題以外にもセキュリティの低下を引き起こす原因となるのも知っていましたか？

HTMLメールでダイレクト・メールがとどいたケースを考えてみましょう。このようなメールの多くは，HTMLファイルのみが添付されていて，そのファイルを開けば，メールの送り主が指定したサーバから画像などを得るしくみとなっています。

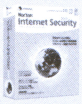
じつは，この動作は「ホームページを見る」という動作と変わりません。しかも，HTMLファイルに仕掛けを作っておくと

- メールを送った相手のうち誰が読んだか
- メールを読んだ時間は
- メールの中でその人が何の情報に興味を持ったか

といったことがサーバ側で簡単に集計できるほか，サーバ側でブラウザが保存している「クッキー」と呼ばれる情報を参照すれば，より，あなたの趣味や趣向が把握しやすくなるのです。

このようにHTMLメールを１回開くだけでも，あなたの個人的な情報が流出する機会を作ってしまうのです…。

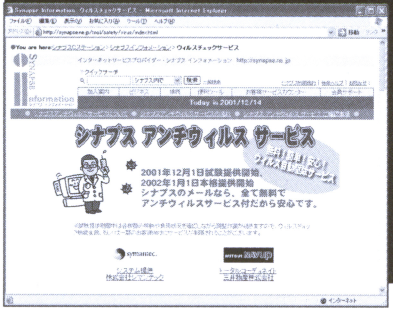
《第２表》発売中のウイルス対策ソフト

商品名	希望小売価格	動作OS	発売元	商品の説明	ホームページ
ウイルスバスター 2002	8,500円	WindowsXp/Me/98/98SE/95/NT4.0/2000	トレンドマイクロ	パーソナルファイアーウォール機能を搭載し，PDA版を同梱した多機能ウイルス対策ソフト	http://www.trendmicro.co.jp/
 2002Norton Internet Security	9,800円	WindowsXp/Me/98/98SE/NT4.0/2000	シマンテック	Norton AntiVirus 2002とNorton Personal Firewall 2002のほか，ネットセキュリティ関連ソフトのパック	http://www.symantec.co.jp/
McAfee.com インターネット セキュリティ Pro.	11,800円	WindowsMe/98/98SE/NT4.0/2000	ソースネクスト	ウイルススキャンオンライン／パーソナルファイアウォール／プライバシーサービスの３製品のパック	http://www.sourcenext.com/

その４ プロバイダのサービスを利用【推奨】

コンピュータ・ウイルスの多くが，感染手段に電子メールを利用します。その出入り口となるメールサーバでウイルスをチェックするサービスをプロバイダ各社が導入しはじめています。

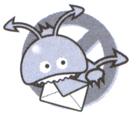
シナプス「アンチウイルスサービス」
http://synapse.ne.jp/tool/safety/virus/



▲SYNAPSEの無料チェックはシマンテックの「Norton Antivirus for Gateways」を利用している

ほとんどのプロバイダが月額500円以下の「付加サービス」として提供していますが，鹿児島でサービスを提供する地方プロバイダのシナプスのように，今後は標準サービスとするプロバイダも増えるかもしれません。

ウイルス対策ソフトと同時に導入すれば，二重のチェック体制を確保できるため非常に効果的です。



ウイルス対策は“危機管理”

日本はコンピュータや高速回線の普及率などを見ても，世界の中でトップクラス。にもかかわらず，その一方では専門家から“セキュリティ認識の甘い国”という指摘があるのも事実です。そんな中で，コンピュータ・ウイルスへの対策は危機管理における一番最初のステップ。あなたのコンピュータがウイルスに感染し，ウイルスメールを他人に送ってしまうだけでも，危機管理能力を疑われてしまうのです。

インターネット時代のセキュリティ対策。BADTRANS.Bを撃退しても，新種のウイルスはこの瞬間にも生まれています。これを機会に，いま一度みんなで考えてみませんか？

「ウイルス対策ソフト」が“駆除”できない理由

「ウイルス対策ソフト」がBADTRANS.Bを検出した場合「駆除できません」という趣旨のメッセージが表示されます。これは，ウイルスが他のプログラムやデータ中に忍び込むのに対し，BADTRANS.Bはワーム活動をする単体のプログラムであるために起こる現象です。

“駆除”はファイル内に潜んでいるところから追い出すことを意味しているためで，エラーが表示されるのは正常な証拠。この場合，該当のプログラムを起動せず，そのまま削除してしまえば問題ありません。

パソコンの心臓部　マイクロプロセッサ誕生30周年企画

CPUの歴史と未来をさぐる！

21世紀でも現役！！老兵は死なず？ 1978年一世を風靡した8ビットCPU「Z80」編

文・絵／くりひろし

なんでもマイコン，入ってます

Dr.D：おはよう，諸君…なんだかだらけているなぁ。いい加減正月も終わって，松も取れたというのに，なんたるちあ！（注：サンタルチアという歌にかけた古いダジャレ）

つぐ美：そんなこと言ったって，ぜんぜん正月過ぎた気がしないんですよ。そもそもお年玉ももらってないワケだし。

Dr.D：いートシしてまだそんなこと言ってるのか…だいたい，技術資料の参照とかいいながらDVDソフト観るのもいい加減にしたら？

つぐ美：ほーい。…って，そういえば，先月号のマイクロプロセッサ誕生30周年企画のなかで，今のテレビとかの家電製品のリモコンやら何やらに，4ビットとか8ビットのマイクロプロセッサが使われてるって言ってたわよね？

Dr.D：お，覚えておったかな？そう，いつもワシが言っておるように，ほとんどあらゆる家電製品の制御部分にはマイクロプロセッサが使われておる！簡単な機能のリモコンなんかには4ビット，テレビ本体や電子レンジ，ビデオなどのちょっと複雑なものには8ビット以上のプロセッサが使われておるのじゃ。

つぐ美：そういえば，ちょっと昔の家電だと「マイコン制御」とかのキャッチフレーズありましたねぇ。

影：あ，知ってる，マイコン炊飯電子ジャーとかあったよね！このマイコンってのは，要するにマイクロコンピュータの略称でしょ？

Dr.D：そう，マイクロプロセッサを中心に，周辺機器との入出力装置を組み合わせた物をマイクロコンピュータと呼ぶのだ！

影：マイコン・ジャーには画面表示としては液晶画面もついてるし，入力のためのボタンもあって，熱源としての出力装置もある！確かに，見かけだけでもコンピュータと言えますね。

Dr.D：いまどきマイコンなしの電化製品（電化って死語？）を探すほうがむずかしいくらいじゃな。電気アンカとかスイッチだけの扇風機とか…で，モノによってはかつてのパソコン市場を席巻したメジャーどころのプロセッサが今でも使われていたりするのじゃよ。

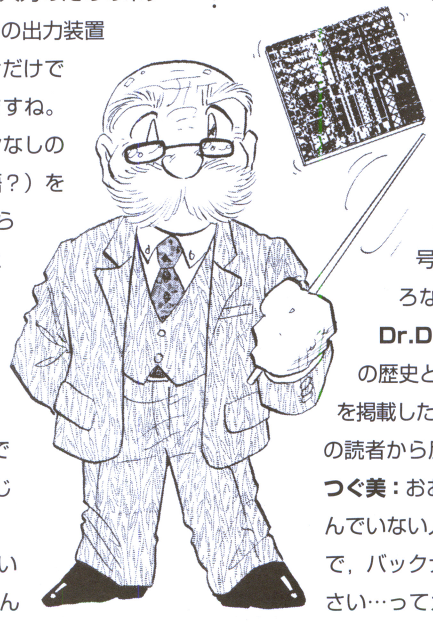
つぐ美：ええ？そんな古いものはもう製造してないんじゃ？

Dr.D：いいモノは永く使われるものなんじゃよ！ワシといっしょでな！

つぐ美：なんだ，ロートルなんだ…。

Dr.D：…だからそういう言いかたは失礼だとゆうておろうがぁ！？

マイクロプロセッサ誕生のおさらい



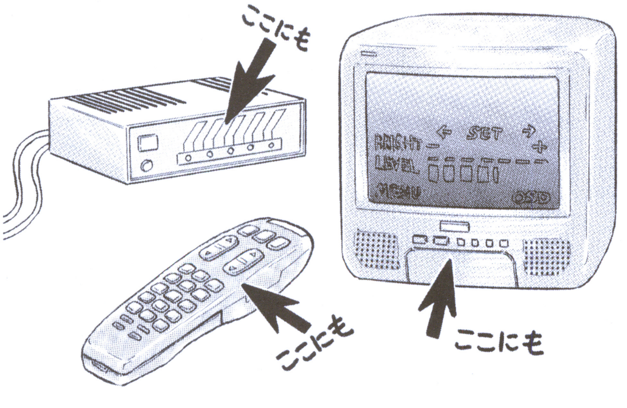
Dr.D：まずはマイクロプロセッサ誕生のおさらいからだ。

つぐ美：8ビット・プロセッサが隆盛を極めるまでの流れをおさえておきたいわね。1月号の記事も見てほしいところなんだけど…。

Dr.D：うむ，1月号で「CPUの歴史と未来をさぐる」って記事を掲載したところ，予想以上に多くの読者から反響があったんじゃな。

つぐ美：おお，それじゃ1月号を読んでいない人は，今すぐ書店に頼んで，バックナンバーを購入してください…ってカンジですね！？

Dr.D：在庫切れになりませんように。



▲現在，多くの電化製品にはマイクロプロセッサが搭載されている。一昔前のパソコンに使われていたプロセッサが使用されている場合もある

マイクロプロセッサは，1969年に日本のビジコン社が，新型電卓を開発する目的で米国のインテル社に新しいICの設計を依頼したことからはじまります。新型の電卓とはいわゆる「プログラム電卓」で，計算式を書き換えることで，いろいろな計算をこなせる，当時としても画期的なアイデアのものでした。

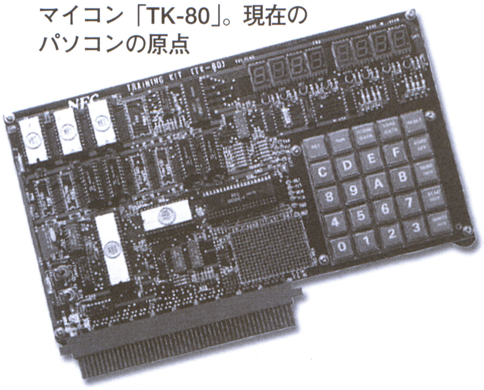
ビジコン社の嶋正利氏（当時）が，インテル社のエンジニアのテッド・ホフ氏と共同で設計を行ったものが，世界初の

4ビット・マイクロプロセッサMCS-4こと，「4004」だったのです。4004は当初複数のカスタムICで実現するはずだった機能を，4ビットの演算回路に複数の汎用レジスタやスタック・レジスタを組み込んで，一つの汎用ICにおさめたものだったのです。

インテル社ではその後，初の8ビットCPU「8008」を独自に開発しましたが，性能は良くありませんでした（電卓向けアーキテクチャそのままに8ビット化したことによる）。そこで改めて嶋氏も加えて，新たな8ビットCPUである「8080」を1973年に開発しました。

8080は，すぐに改良されて「8080A」になりますが，1975年に実質的に世界初のパーソナル・コンピュータ，MITS社Altair8800に採用され，一躍有名になりました。その後，メジャーなPCとしては，1977年にタンディのTRS-80にも採用されました。これも日本でも有名だったパソコンですね。

▶ビジコン社のプログラム電卓（世界初の4ビット・マイクロプロセッサ「4004」を搭載）



▼8080A採用のワンボードマイコン「TK-80」。現在のパソコンの原点

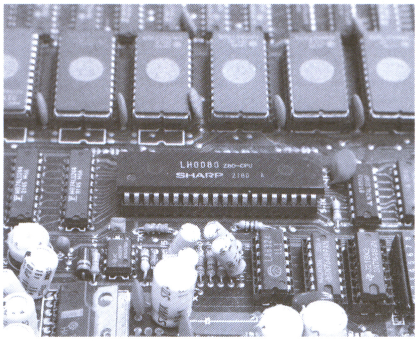
一方，日本では当時の「ワンボードマイコン」と呼ばれたホビー用コンピュータのCPUとして使われました。1976年発売のNECのTK-80にはじまり，たくさんのマイコンが「8080A」を採用していました。

日本の8080A採用機種

メーカー	機種名
NEC	TK-80（厳密にはμPD8080Aを搭載）
	TK-80E（廉価版）
	TK-80BS（BASIC言語搭載型）
	COMPO-80（TK-80完成品）
東芝	EX-80（TVインターフェース内蔵）
	EX-80/BS（BASIC言語搭載型）
ロジックシステムズ	MP-80
シャープ	SHB80T

その他のワンボードマイコン

メーカー	機種名	採用プロセッサ
コモドール	KIM-1	6502
富士通	LKIT-8	6800(MB8861)
パナファコム	LKIT-16	MN1610（16ビット！）
日立	H68/TR	6800



▲これが「Z80」CPU。1978年以降1980年代の主要パソコンはZ80が搭載された

つぐ美：このころのワンボードマイコンってパソコンの原点なんですねぇ。

Dr.D：いまから思えば凶悪なコンピュータじゃったなぁ。そもそも，現在の読者がアレをコンピュータと思えるかどうか？入力は英文字A～Fを含む16進数対応（というか，それ以外は若干のファンクションキーがあるだけ）のキーボード，出力は7セグメントLEDの8ケタ表示で，ようやく16進数を表すことができるって程度。要するに，プログラミングなんて16進数のマシン語でそのまま打ち込むってことだ。

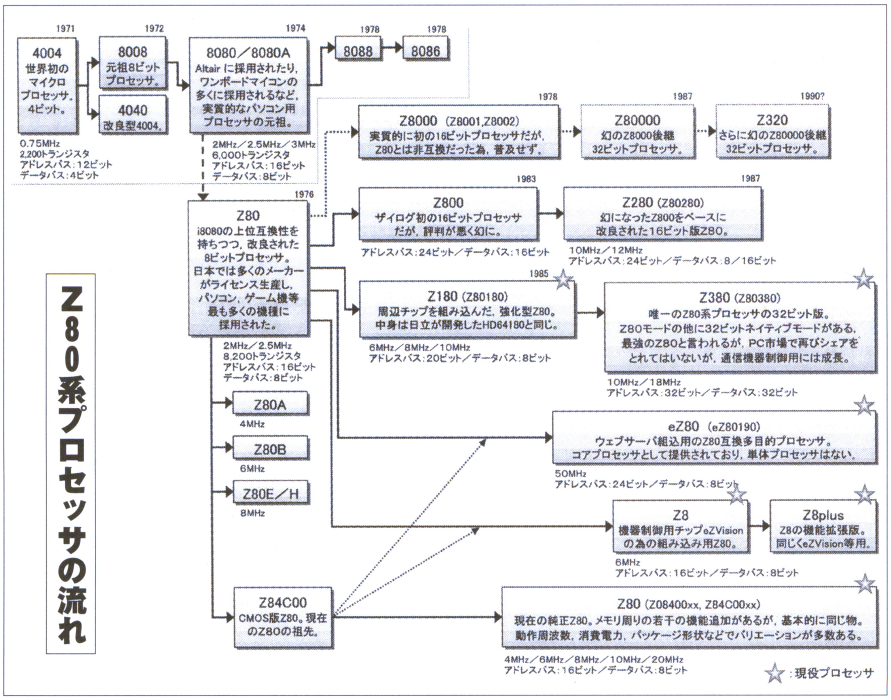
つぐ美：伝説のAltair8800の，トグル・スイッチで2進数をパチパチ（ってスイッチのon/offのことよ）しながらプログラミングするよりはマシよねぇ？

Dr.D：さぁ？ま，それでも当時のファンはゲームを作ったりしとったよ。電卓ゲームみたいな奴じゃな。

つぐ美：ゲーム・ウォッチみたいなの？

Dr.D：グラフィックが出るワケじゃないからのう…ま，一昔前に流行ったポケベル・メッセージのようなもんだと言えば，つぐ美くんならわかるじゃろ？

つぐ美：いやーんアタシってばケータイ世代だからわかんなーい(^o^)



Z80登場

さて，8080Aを開発した嶋氏は，4004を開発したときのインテル社メンバーが独立して作った新しい会社，ザイログ社に招かれて新しい8ビット・プロセッサを開発することになります。これが「Z80」で，1976年に誕生しました。

Z80は8080Aの改良を主眼に，多くのアーキテクチャの改善が図られています。後の8086系16ビット・プロセッサの祖先は，8080AよりむしろZ80と言えるかもしれません。

8080A→Z80への改良点

- ・命令セットの拡充：8080Aは全部で78個の命令セットを持っていたが，Z80ではさらに独自の80個の命令を追加して，全部で158個に倍増させた。すなわち，8080Aの上位互換プロセッサとなる。
- ・命令記述の改善：8080Aよりもわかりやすいマシン語記述ができるようになった。
- ・メモリコントロール：メモリ(DRAM)アクセスのリフレッシュを行うインターフェースを持たせた。
- ・レジスタ拡充：多数多種のレジスタを増設。はじめてインデックス・レジスタを持ち，より複雑なプログラミングが可能になった。
- ・インタラプト機能強化：インタラプト用メモリテーブルを持ち，それまで分岐処理でごまかしていた割り込み処理を本格的に可能とした。
- ・電源の改良：8080Aが3電源(＋5V，－5V，＋12V)を必要としたところを，＋5Vの単一電源化し使いやすくした。

さて，“良いモノは売れる”の法則どおりに，8080Aとコード互換であるうえに，より高性能，高機能，使いやすい，プログラムしやすい，ときては売れないはずがありません。本来8080Aが適用されてもおかしくなかった本格的な8ビット・コンピュータ時代は，あまねくZ80が適用されることになるのです。

ホントウに人気が高かったZ80

Dr.D：で，それはもうホントウにZ80は広まりまくったワケだ。

影：Z80搭載じゃない8ビット・パソコンって，結局のところ，「6502」のアップルⅡ系か，「6809」のFMシリーズくらいなものでしょ？さもなきゃ16ビット化して「8086」系か「68000」系に移行しちゃうワケで。

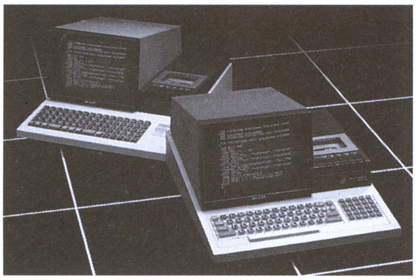
Dr.D：おお，よくわかっとるじゃないか。

影：ふふ，ダテにパソ破正当伝承者を名乗ってはおりませんって。

Dr.D：なるほど，パソコンの中身についでは誰よりもよく知っておるのも道理…って，やめんか～い！

日本ではまず，1978年にシャープの「MZ-80K」に採用されたのを皮切りに，翌年NECの「PC-8001」に採用され，1981年に「6809」を採用した富士通「FM-8」が出るまではZ80の独壇場が続くのです。

Z80を使ったパソコンでは，当初PC-8001とMZ-80の二大勢力ができあがっており，そこにFMシリーズが加わって，80年代前半はシャープ・富士通・NECのパソコン三国志時代が繰り広げられるのです。



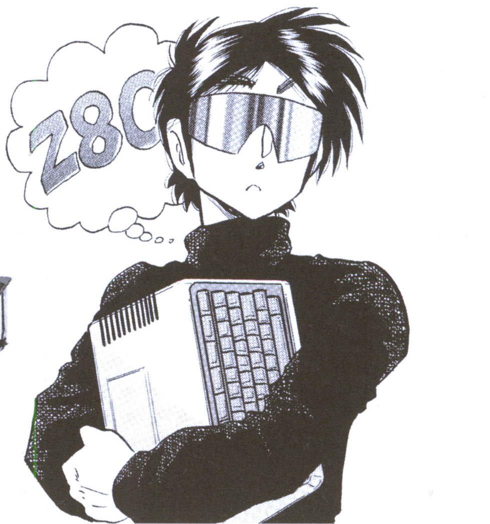
▲Z80搭載第1号パソコン「MZ-80K」(シャープ)。データ記録のためのカセットテープ・レコーダが内蔵されていた

▲NECの「PC-8001」。マイコンブームの火付け役となった

▲▶1983年に登場したNECの「PC-8801mkⅡ」(上)には5インチFDDが，「PC-6601」(右)には3.5インチFDDが搭載され，この年からFDD搭載パソコンが増えはじめた

Z80搭載パソコン一覧（除くMSX系）

年	メーカー	機種
1978	シャープ	MZ-80K
1979	NEC	PC-8001
1980	ソード	M203markV
	シャープ	MZ-80C
	シャープ	MZ-80K2
1981	シャープ	MZ80B
	シャープ	MZ-80K2E
	NEC	PC-8801
	NEC	PC-6001
1982	東芝	PASOPIA
	シャープ	MZ-200
	シャープ	MZ-700
	シャープ	MZ-1200
	シャープ	X1
	シンクレア	ZX-81
	SONY	SMC-70
	SANYO	PHC-25
1983	NEC	PC-6601
	NEC	PC-8001mkⅡ
	NEC	PC-8801mkⅡ
	NEC	PC-8801mkⅡSR
	シャープ	MZ-2200
	シャープ	X1C
	シャープ	X1D
	東芝	PASOPIA7
	SONY	SMC-70G
	SONY	SMC-777
	EPSON	QC-10
	三菱	MULTI8
	カシオ	FP-1000
	カシオ	FP-1100
	カシオ	FP-5700
	カシオ	PV-2000
	ソード	M5
	バンダイ	RX-78GUNDAM
	セガ	SC-3000
	セガ	SC-3000H
1984	NEC	PC-6001mkⅡ
	NEC	PC-6001mkⅡSR
	NEC	PC-6601SR
	NEC	PC-8001mkⅡSR
	シャープ	MZ-1500
	東芝	PASOPIA5
	EPSON	HC-40
	EPSON	HC-88
1985	シャープ	X1turbo
	ソード	M5jr
1986	シャープ	X1G
	シャープ	X1turboⅡ
	シャープ	X1turboⅢ
	シャープ	X1turboZ
1987	シャープ	X1twin
	シャープ	X1turboZⅡ



Dr.D：ま，多少パソコンとは呼びにくいものも含まれとるが，プログラミング可能なZ80搭載コンピュータということで並べてみたぞ。

つぐ美：おお，ガンダム！なつかしいですねぇ！これもZ80だったのかぁ…よくこんな名前つけたなぁ。

影：ふっ，認めたくないものだなぁ…自分自身の若さ故の過ちというものを…

Dr.D：ザクとはちがうのだよ，ザクとはぁ！

つぐ美：…連邦VSジオンDXやってるわね？

Z80バリエーション

Z80にも，動作周波数のちがいや，互換プロセッサ，さらにその改良型などいろいろなバリエーションがあります。Z80は一時期あまりにも製造するメーカーが多かったため，細かなちがいを含めると，当時のものだけで十数種類はあったと思われます。

主な初代Z80のバリエーション／互換プロセッサ

形式名	メーカー	特徴
Z80	ザイログ	2MHz/2.5MHz
Z80A	ザイログ	4MHz
Z80B	ザイログ	6MHz
Z80E	ザイログ	8MHz
Z80H	ザイログ	8MHz
Z84C00	ザイログ	CMOS採用による低消費電力型
μPD780	NEC	Z80互換
LH0080	シャープ	Z80互換
BUI8400	ローム	Z80互換
MK3880	MOSTEK	Z80互換
HD64180	日立	周辺チップ組み込み型
R800	松下	28MHz動作のできる高速型
KC80	川崎製鉄	RISCアーキテクチャ採用

なんでもZ80

影：Z80ってさ，当時のアーケード・ゲーム機にもたくさん使われていたんだって。

Dr.D：うむ，ゲーム・センターの専用ゲーム機といっても，ソフト開発にはプログラミングが必要じゃろ？当時Z80は手に入りやすく，しかもプログラミング環境も整っていて，参考書も多いときては，自然と使われるわけじゃ。もちろん，互換プロセッサも相当作られていたと思うぞ。大流行したインベーダー・ゲームなんてのは，プロセッサどころか，ゲーム自体が互換みたいなもんで…。

つぐ美：いや，そっちは海賊版でしょ！

影：あのゼビウスのゲーム基板にもZ80が使われていたという話だけど，8ビット・パソコンじゃマトモなゼビウスはゲームになんなかったよ？

Dr.D：そりゃ，ゲーム基板とパソコンを一緒にしてもらっちゃ困るな。画面表示部分はグラフィック表示に向いたモノを使っておるし（パソコンは文字表示に向いた物を当時は使っていた），このゼビウスではZ80は3個差しなのだ！

影：ほえー，大盤ぶるまいですな。

Z80の絶頂期

さて，日本の8ビット・パソコン時代を引っ張ってきたZ80ですが，もう一花咲かせることになります。1983年にアスキーが提唱した統一規格パソコンであるMSX（なんでもマイクロソフト無限大という意味なんだそうな）のプロセッサとして採用されるのでした。

国内14社をはじめ，確認されているだけで海外も含め20社以上がMSXパソコンを販売しました。各社3～5モデルは出していましたから，初代MSXだけでも80機種以上（筆者が数えた限りでは85機種…）ありました。結果として，メーカー製パソコンに関しては，Z80は日本で最も採用機種の数が多かった（台数ではなく）プロセッサになるのです。

その後，1985年にMSX2（このころが最もMSXが隆盛を極めた），1986年にMSX2＋，最終的に1990年のMSXturbo

◀松下電器のMSXパソコン「CF-2000」とその周辺機器

MSXの基本仕様

CPU：	Z80A
メモリサイズ：	機種によるが8～64KB（最大1MB可能）
グラフィック：	256×192ドット16色表示
サウンド：	8オクターブ3重和音
OS：	MSX-BASIC（Microsoft拡張BASIC V4.5），MSX-DOS(別売) MSX規格ソフトウエアROMカートリッジ（ゲーム・カートリッジなど）
データ入出力：	1200／2400ボーのカセット・データレコーダ
特徴：	MSX規格パソコンは各社でソフトの互換性がある

MSX機種一覧（一部）

メーカー名	形式	名称	特徴
CANON	V-10	V-10	スイッチオン
SONY	HB55	HIT BIT	人々のヒットビット
	HB75	HIT BIT	HB55上位機種
	HB101	HIT BIT MEZZO	デザイン一新
TOSHIBA	HX10D	PASOPIA IQ	カートリッジポン
YAMAHA	YIS503	YIS503	作曲機能
	YIS303	YIS303	YIS303上位機種
PIONEER	PX-7	"Palcom"	家の中が面白い
日立	MB-H1,MBH1E	H1	
	MB-H2,MB-H21,MB-H25	H2	
富士通	MB25150	FM-X	友達パソコン
松下電器	CF2000	MSX2000	夢のパソコン
サンヨー	MPC2	WAVY	ライトペンMSX
カシオ	PV7	PV-7	超低価格(19,800円)
	PV16	PV-16	

Rまで多くのメーカーがZ80を採用し続けたのです。ちなみに，MSXturboRでは厳密にはZ80ではなく，松下電器が開発した28MHz動作可能なZ80互換プロセッサであるR800が採用されていました。

しかし，MSXが発表になった時点で，既にビジネス系は16ビット化がはじまっていました。ゲーム機としても，ちょうど同時期（1983年）にファミリーコンピュータが発売になっており，性能的にどっちつかずのMSXはやがてPC市場からは姿を消すことになります。MSXそのものの志は高かったので，今でも熱心なMSXファンがたくさんいます。

インターネットで検索すれば，いくつものMSXのファンサイトを発見できるでしょう。現在では機体そのものが入手しづらくなっているので，MSX用のエミュレータもあるようです。

表舞台から消えていくZ80

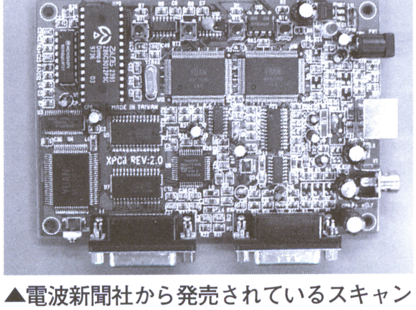
つぐ美：それにしても，どうしていまじゃパソコンにザイログ社…というか，Z80系のプロセッサは使われていないのかしら？

Dr.D：いろいろ理由はあるのだが，互換性問題などでやはり16ビット化の波に乗れなかったのがイタイな。まぁ，IBM-PCの成功には誰も勝てなかったのだが…

嶋氏もたずさわって，1978年にザイログ社で最初に開発された16ビット・プロセッサ「Z8000」は，8086よりやや遅れたものの，1979年には出荷されました。しかし，Z80とはコード非互換であったためにほとんど普及しませんでした。その後，ザイログ社独力で「Z800」を開発しましたが，1983年までかかったこともあってか，なぜか評判が悪く，これも幻に終わります。結局，本格的な16ビット・プロセッサは1987年にようやく「Z280」として完成することになります。

しかし，時は既に32ビット・プロセッサの時代に突入していました。ザイログも1987年に「Z80000」，その後「Z320」と32ビット・プロセッサを開発しましたが，パソコン市場は既に大半がインテル系プロセッサ，何割かが68000系プロセッサで大勢は占められてしまっていたのです。

こうして「パソコンのCPU市場」からは，ザイログ系プロセッサは姿を消していったのですが…。

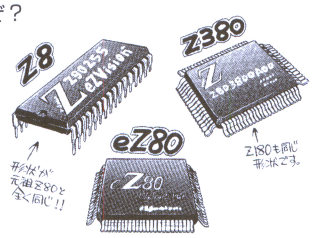


▲電波新聞社から発売されているスキャンコンバータ「XPC2」には「Z80」から派生した「Z8」を使った（いわば親戚）ザイログ社の「Z86C30」が！

Dr.D：Z800というと，まず思い出すのが往年の名機，フジの8ミリカメラZ800だなぁ。

つぐ美：なにそれ，古すぎない！？あたしだったら，シャープのザウルスの前身である電子手帳WIZ(Z800)を思い出すわぁ。

影：いやぁ，それはどっちも五十歩百歩だと思うぞ？

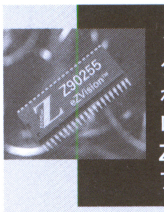


どっこい生きてたZ80

…で，Z80は生き残っていました。Z80そのものがあまりにも広まったこと，CPUが16ビット／32ビット時代になっても周辺機器などは案外8ビットで用が済むものが多かったため，Z80は21世紀になってもしっかり生きています。

ザイログ社の製品分類によれば，マイクロプロセッサ扱いのものと，制御プロセッサ扱いのものがありますが，いずれも用途をちがえているだけで，基本的にはZ80の後継プロセッサまたはZ80そのものをコア（プロセッサの核の部分）に使っているものです。

たとえば，Z8とかZ8plusと呼ばれるZ80アーキテクチャのコアは，「eZ Vision」なる組み込み型プロセッサに使われています。eZVisionはケーブルテレビのSTB（セット・トップ・ボックス）やビデオデッキ，テレビ，それらの多機能リモコンのプロセッサとして使われており，OSDと呼ばれる画面表示機能によって10×24文字の設定画面をテレビに表示させることができるのです（読者諸君もチャンネル調整などでテレビにオンスクリーン表示のメニューを見たことがあるはず）。



◀セット・トップ・ボックスやビデオデッキに使われている組み込み型プロセッサ「eZVision」。Z80アーキテクチャのコアを使用している

ザイログ社のホームページ＝
http://www.zilog.com/

また，「eZ80」というZ80コンパチの高性能プロセッサは，ウェブサーバ構築用（IDC：インターネット・データ・センターなどで，大量のウェブ・アクセスに対しての負荷分散を行ったりするために，小型のサーバを多数設置してクラスター型のサーバを構築することがある）の組込用多目的マイクロプロセッサです。

Z80互換は当然ですが，16MB空間のリニアアドレッシングが可能な他，同じ動作周波数ならば4倍のスピードで動作できます。すなわち初代Z80の100倍の性能を持っていることになるのです。

さらに，幻だったZ8000なども周辺チップの「Z8530」が今でもシリアル・インターフェースの制御用プロセッサとして使われていたりもします。

Z80とその仲間は，パソコンのCPUとしてこそ使われなくなりましたが，インターネット時代の通信機器制御，家電AV製品の制御用プロセッサとして現在も活躍しているのです。

影：Z8plusってゆーと，センチネルに出てきたZガンダムの量産型の…。

つぐ美：わかんないって。

影：eZ80ってゆーと，08小隊の陸戦型ガンダムの改造型の…。

つぐ美：だからいーかげんにガンダムネタから離れなさいって。

Dr.D：なんだかRX-78といいザイログ社ネタなのにガンダムネタが多いなぁ…って，ZってジオンのZかい？

つぐ美：Dr.Dまでそのネタかい〜〜〜！

※Z86C30系は多目的マイクロコントローラに分類されており，Z8（Z80じゃなくて）CPUの他にRAM，ROM，監視タイマやカウンタを集めてワンチップマイコン化したもの

■連載■ PCハードウエアテクニック

市販アップグレード・キットで旧型FMVをWindows XP対応化

旧型モデルDOS/V機 アップグレード講座

パソコンの機能や操作性の向上に欠かせないＯＳのバージョンアップですが，新しいＯＳの利用には，ハードウエアのアップグレードが欠かせません。昨年登場したWindows XPも例外ではありません。そこでお勧めしたいのが，ペンティアムMMXやペンティアムⅡなどの旧モデルマシンのアップグレードです。市販のアップグレード・キットを旧型マシンに組み込むことで，押入に眠っているパソコンを現役バリバリのWindows XPマシンに再生することも簡単です。

1 アップグレード・キットでお手軽パワーアップ

大幅な機能の充実や動作の安定化がはかられたWindows XPは，ブロードバンドネット利用をはじめ，動画・サウンドなどが扱えるマルチメディア対応のパソコンには，欠かせないパワフルな最新ＯＳです。その一方，Windows XPの導入には，750MHz以上のCPU（500MHz以上のCPUが必要），128ＭＢ以上のメモリ，300ＭＢ以上のハードディスク（ＨＤＤ）の空き容量が必要で，従来のWindows ＯＳよりもワンランク上の性能のマシンが，要求されます。とはいえ，最新モデルの新規購入には，最低でも10万円の予算はかかります。これまで使ってきた旧型マシンでWindows XPが使えれば，と思うのは無理のない話ですね。

このような場合は，新しいＣＰＵやマザーボードに載せ替え，マシンのアップグレードを行ってみてはいかがでしょう。手軽に手持ちのパソコンをWindows XP対応マシンに変身できるのが，今回取り上げた若松通商から発売の「アップグレード・キット」です。

このキットは，Windows XP HomeEdition（ＯＥＭ版）またはWindows XP Professional（ＯＥＭ版）とマザーボードをセットにした製品で，ＣＰＵとメモリモジュール，ＨＤＤをプラスすることで，旧型モデルのアップグレードを行うことが可能です。

なお，このキットを利用しない場合でも組み込み方法はほとんど同じなので，各ＤＯＳ／Ｖパーツを独自に購入しアップグレードを予定している場合にも役立つはずです。

交換に最適なベビーサイズの多機能マザーボード

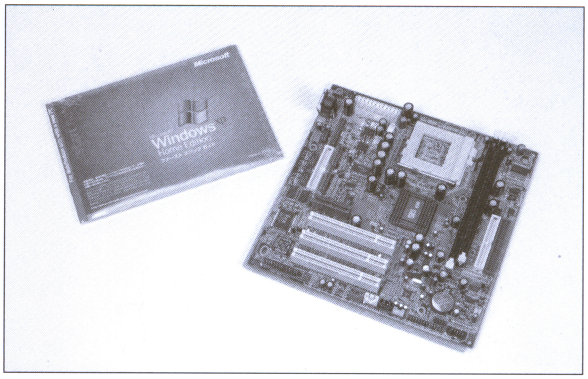
本キットの中心になるのが，ZIDA

若松通商のWindowsXP対応アップグレード・キット

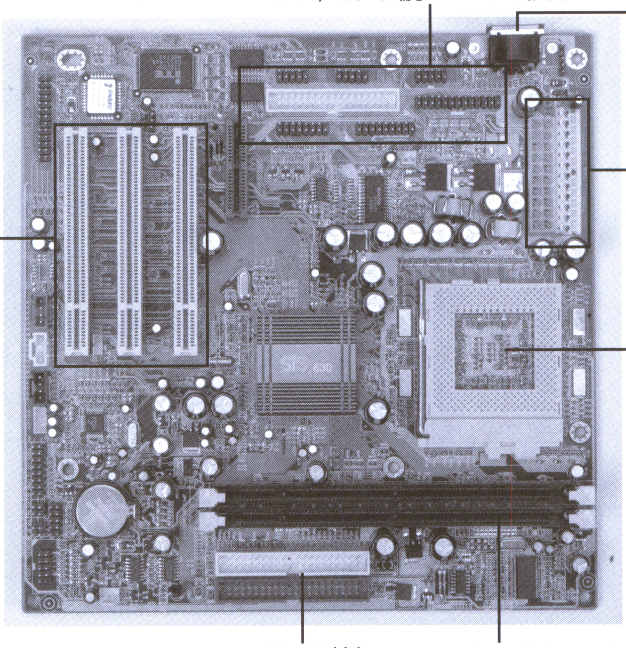
マザーボード「V630E」仕様

サポートCPU	Intel Pentium III FCPGA, Celeron PPGA or cyrix III(Socket 370)
FSB	66/100/133MHz
メモリスロット	DIMM ×2(合計2GBまで)100/133MHz SDRAM DIMM
チップセット	SiS630E
スロット・ポート	PCI ×3,USB ×4, IrDAポート×1, FDD ×1,LPTポート×1,COMポート×2
IDE	Ultra-DMA33/66 IDEデバイス×4
ネットワーク	LANオンボード(Wake on LAN対応)
ビデオ	ビデオオンボードAGP Controller inside SiS 630E Chipset AGP v2.0 Compliant
サウンド	Integrated AC97 サウンドオンボード AC97 2.1 Compliant Codec with 3D Stereo Enhancement Line-out ×1,Line-in ×1,Mic-in ×1,CD-in ×1,AUX-in ×1,Game Port ×1
電源	AT& ATX電源対応
サイズ	Baby ATホームファクタ,1220mm x 220mm

アップグレード・キットには，マザーボード「V630E」とWindows XP Home Edition（ＯＥＭ版）のセット（21,780円）とWindows XP Professional（ＯＥＭ版）セット（27,780円）の２タイプがある。



アップグレード・キット　マザーボード

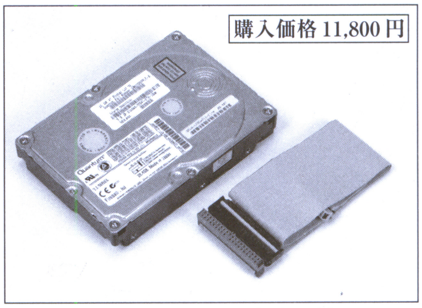


▲アップグレード・キットのマザーボードは，ベビーATフォームファクタの多機能タイプ「V630E」を採用。一般的な旧型市販PCであれば，交換可能だ

OS － Windows XP －

▲アップグレード・キットは，最新OSのWindows XP（OEM版）が同梱

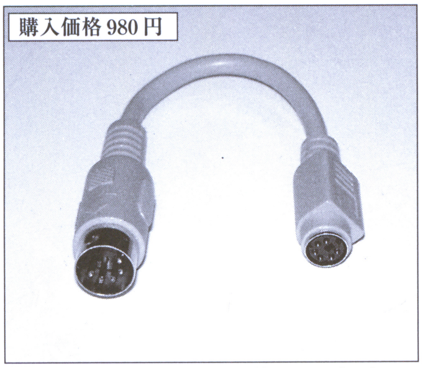
ハードディスク／別途購入



▲40GBクラスなら，安くて使いよい。動画ファイルなどを扱う場合も十分だ。ケーブルは，マザーボードに添付

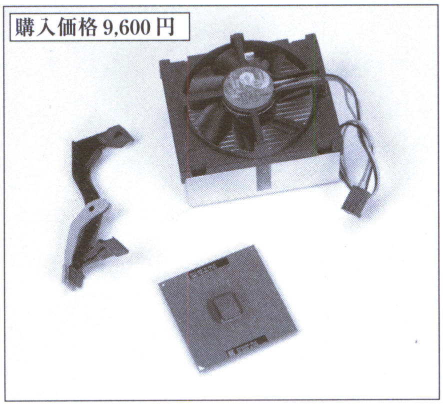
※価格はいずれも秋葉原店頭調べ（2001年末時点）

PS/2キーボード変換アダプタ／別途購入



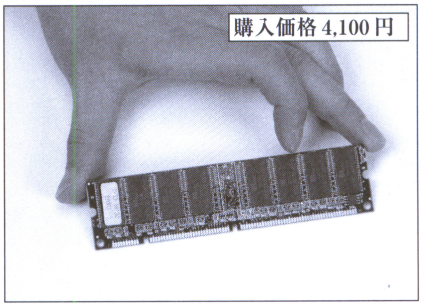
▲PS/2キーボードを利用する場合は，DIN－PS/2変換アダプタを用意

CPU&冷却ファン／別途購入



▲CPUは，セレロン1GHzを用意。リテール版には専用冷却ファンが付属

メモリ・モジュール／別途購入



▲安価なSDRAMメモリが使える。できれば256MBタイプを選んでおこう

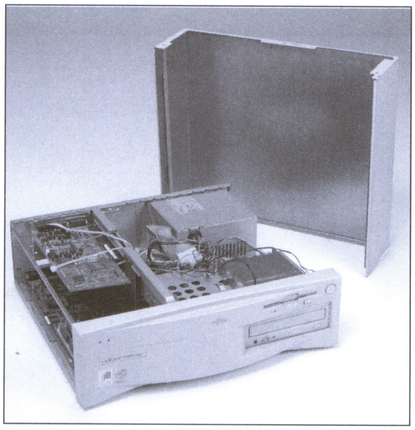
TECHNOLOGY社製「V630E」マザーボードです。このマザーは，ペンティアムⅢ，セレロン，サイリクスⅢのソケット370ＣＰＵに対応，オンボードでビデオやサウンド，ネットワーク機能まで搭載した多機能マザーです。

一般的なＤＯＳ／Ｖ自作機のＡＴＸマザーよりも，ひとまわり小さめなベビーＡＴホームファクタのため，タワータイプやデスクトップタイプなどの市販パソコンの交換用マザーにも最適です。また，旧型モデルに多いＡＴ仕様なので，電源ユニットやキーボード（ＤＩＮ端子タイプ）をそのまま利用できます。電源は，ＡＴ用端子のほかにＡＴＸ用もあり，ほとんどのＤＯＳ／Ｖ機に対応可能です。もちろんＡＴタイプの自作ＤＯＳ／Ｖ機用にも使えます。

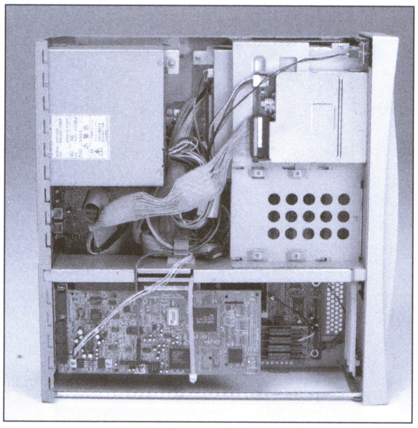
ほかのマザーを購入する場合は，このようなベビーＡＴマザーを選択すれば安心です。

マザーボードは，パソコン全体の機能や性能を決めてしまうほど重要なパーツです。対応ＣＰＵやインターフェースの種類など，事前にチェックしておきたいものです。市販パソコンは，メーカー独自のマザーボードを採用している場合がほとんどで，穴の位置が異なりＰＣＩやＩＳＡなどの拡張カードが，マザーの拡張スロットから

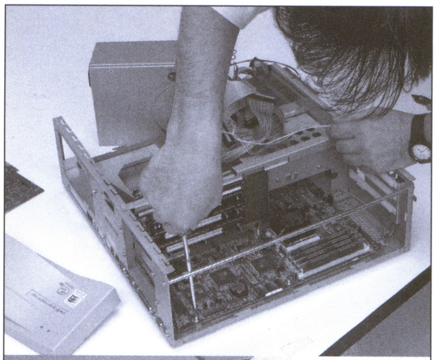
組み立て手順を忘れずに注意深くパーツを取り外そう!! 旧モデルパソコンの分解



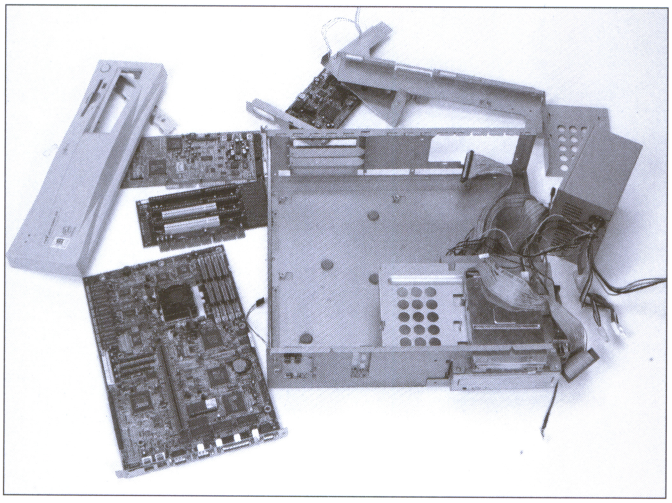
▲今回，ターゲットにした旧型市販PCは，ペンティアムMMX搭載の富士通FMVシリーズ



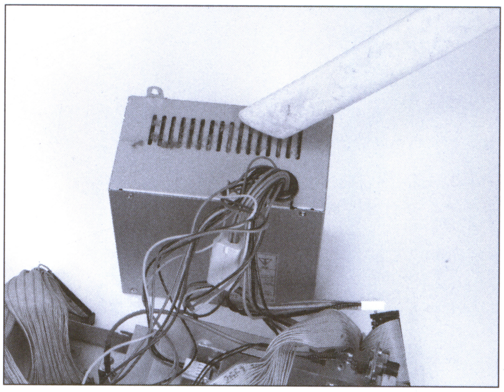
▲マザーボードは独自設計で，そのままの状態でのアップグレードは極めて難しいようだ



▶PC内部のパーツはすべて慎重に取り外す。組み立てに備え取り外した順序は覚えておこう



▲分解が終了したFMV。ケース，フロッピーディスク・ドライブ，CD-ROMドライブ，電源ユニットはそのまま流用する



▶分解と同時に、ケース内部にたまったホコリは掃除機などを使ってきれいに除いておこう

使えない場合が多いようです。この理由から，交換用マザーの条件として，パソコン利用に不可欠な基本的なインターフェースは，オンボードでサポートしているものがベストです。

その点，「V630E」はビデオやLAN（10/100BASE-TX），サウンド入出力をオンボードで対応しているほか，プリンタ・ポート，シリアル（COM）ポート，USBポートも含め，すべてケーブル接続で端子をパソコンのリアパネルに取り付けることができます。市販パソコンへのマザー交換でトラブることの多い端子位置の違いもこれで解消できるわけです。

CPU，メモリ，HDDのベストな選択基準

本キットでは，マザーボード，OS以外は，別途用意することになりますが，今回はWindows XPの導入が目的なので，性能に余裕のあるものを選びました。CPUにセレロン1GHz，HDDは40GB，メモリには128MBというラインナップです。

CPUは，ペンティアムIIIという選択肢もありますが，コストパフォーマンスの点でセレロンにしましたが，1GHzクラスなら，Windows XPでも安心できます。最近，低価格化が進んでいるHDDには，1万円前後で購入できる40GBクラスで大丈夫です。大容量ファイルを扱うビデオ編集などの用途にも，40GBあれば十分でしょう。

少し心配なのがメモリです。メモリを多量に消費するWindows XPでは，256MB以上は確保したいところです。今回のマザーボードは，2基のSDRAMメモリスロットがあるので，アップグレード後に拡張しておきましょう。メモリ容量を増やすことで，スピード・アップにも貢献できます。

2 交換マザーボードへのCPU，メモリの取り付け

パソコン・アップグレードへの第一歩は，マザーボードのセットアップです。解体した旧型マシンのマザーボードと交換する前に，あらかじめアップグレード用マザーボードにＣＰＵや冷却ファン，メモリの各パーツを取り付けることから始めます。

とはいえ，一つずつのパーツの極性や方向に注意を払いながら確実に取り付けていくことは，ＤＯＳ／Ｖ機の自作の場合と何らかわるところはありません。

トラブルの多いCPUの装着は要注意!!

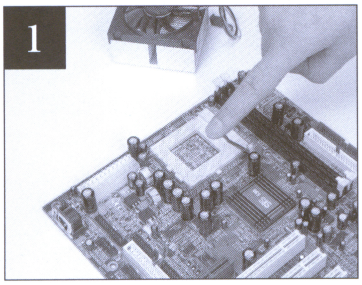
ＤＯＳ／Ｖ機製作に経験のある人であれば，目をつぶっていてもできるよ…といわれそうですが，じつはトラブルの多くが，マザーボードへのパーツの取り付け不備が原因になることが多いのです。

とくに，ＣＰＵのソケット装着や冷却フィンの設置には要注意です。ピン数が多く（Socket 370では，370本），発熱量の多いＣＰＵは，いい加減な方法で取り付けると，動作しないどころか最悪ＣＰＵ自体を破壊しかねません。

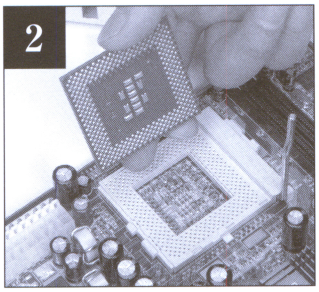
ＣＰＵ取り付けのポイントは，ピンの方向を合わせしっかりソケットに差し込み，ロックレバーで確実に固定することと，ＣＰＵと冷却フィンを隙間なく密着させることの２点です。

また，ピン方向を間違えたまま無理矢理ソケットに押し込もうとすると，ピンが折れてしまうこともあります。こうな

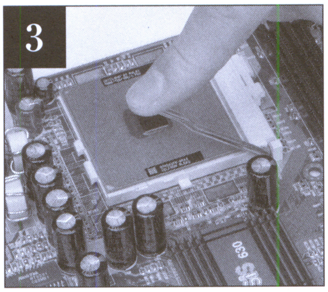
マザーボードのセットアップはパーツの方向，極性に注意を払って確実に行おう!!



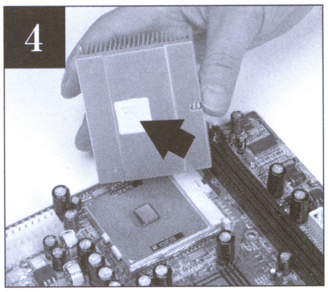
▲「V630E」マザーのCPUソケットは，Socket370タイプ。ここにセレロンCPUをセット



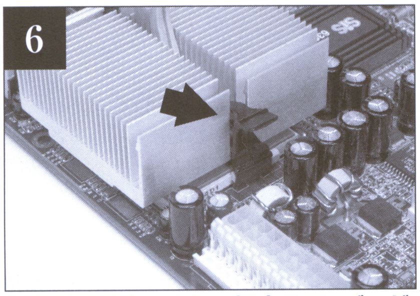
▲ロックレバーを上げ，CPUとソケットのピン配置を確認し確実に装着



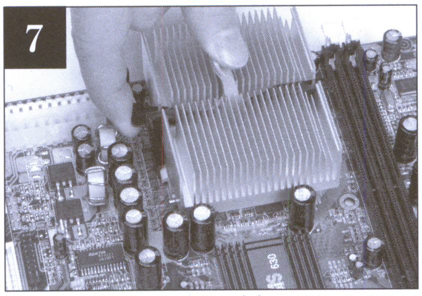
▲ソケット装着後は，ロックレバーを倒し，CPUをロックしよう



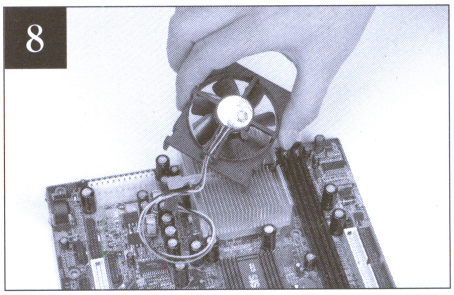
▲冷却フィンのCPU接触部分にシリコングリスまたはシートを貼り付け，廃熱効果を高める



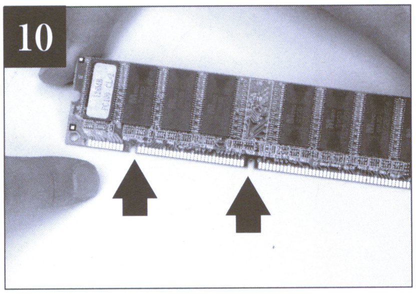
▲上手く固定できない場合は，レバーがフックに掛かっているかを確認



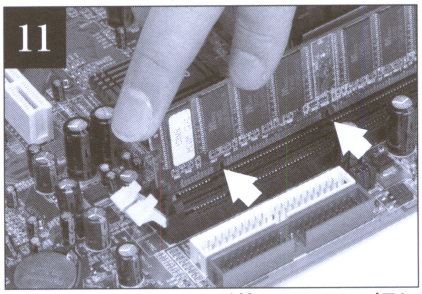
▲テンションレバーを倒すことでCPUに冷却フィンが密着する



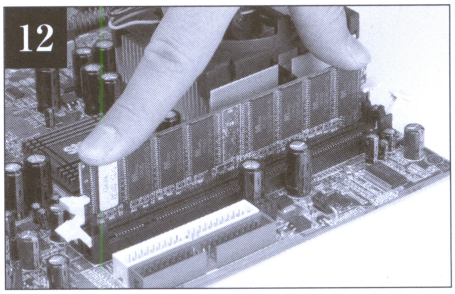
▲冷却フィンの上にクーリングファンを取り付ける



▲SDRAMメモリには逆差し防止用に中央の他にもう一つ切り欠きがある



▲メモリスロットの形状にメモリの切り欠き部分を合わせる



▲スロットへの装着はメモリモジュールの左右へ均等に力を加えながら押し込むこと

ると，修復することは無理です。ソケット側とＣＰＵ側のピンの配置をあらかじめ確認して装着するようにしましょう。

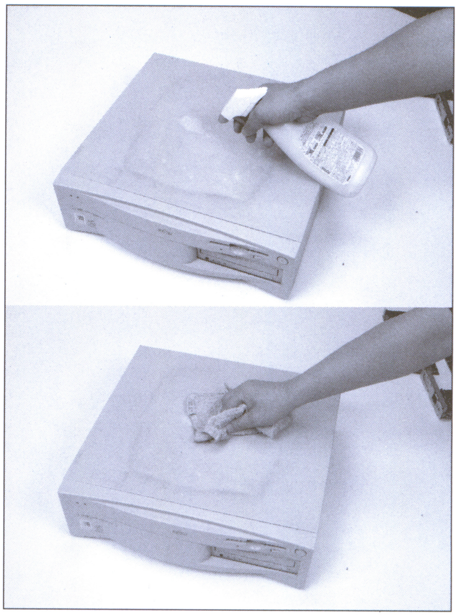
今回のように1 GHzクラスのＣＰＵでは，発熱量も膨大で冷却フィンとファンなしには，瞬時に破壊してしまいます。隙間なく効率よく廃熱を行うためには，シリコングリスやシートで密着することです。これらはＣＰＵに付属している場合もありますが，見あたらないときはできるだけ良質なものを購入しましょう。シリコングリスは，接触部分からはみ出さないように，適度な量を塗ってください。グリスの分量が多すぎると，ＣＰＵの胴体部分に流れショートなどのトラブルのもとになります。はみ出たグリスは布やティッシュペーパーなどで，きれいにふき取っていきます。

よくある事故が，冷却フィン取り付け時に，過剰な力がＣＰＵにかかり，端が欠損してしまう，いわゆるＣＰＵの「コア欠け」が起きてしまいます。このコア欠けはＡＭＤ系ＣＰＵに多いようですが，テンションレバーで強く閉めるため，フィン位置が多少でもずれているとＣＰＵに偏った力がかかることがその原因です。冷却ファンに付属の図解図などで確認し，正確に取り付けを行ってください。

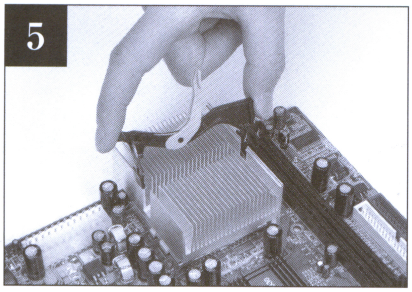
本アップグレード・キットのマザーボード「V630E」は，ＣＰＵソケットにSocket 370が採用されています。このタイプは，フィン取り付けにソケット両サイドのフックにテンションレバーを引っかけ，そのうえでレバーを倒すことでＣＰＵとの密着度を高めるようになっています。そのため，他のタイプのソケットやスロットよりも，余計な力を加えずに簡単に取り付けができるので，初心者にも安心です。冷却フィンとファンの取り付け後は，ファン電源のケーブル接続を忘れずにしておきましょう。

安価なSDRAMメモリが使える「V630E」マザー

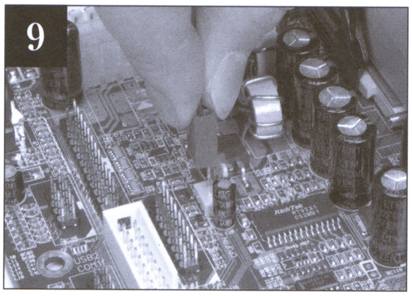
ＣＰＵの次は，メモリモジュールの装着です。ところで現在，ＤＯＳ／Ｖ機向けのメモリには，いくつかの種類がありますが，本キットの「V630E」マザーは，SDRAMをサポートしています。ペンティアム4やアスロンＸＰなどの最新ＣＰＵでは，高速性を追求するために高価なRDRAMや従来の2倍のスピードで読み書きできるDDR SDRAMなどが主流になりつつあります。これらのメモリに比べるとデータ転送速度は劣るものの価格の面でメリットの多いSDRAMは，安価に効果的なアップグレードを行う最適なメモリといえるでしょう。



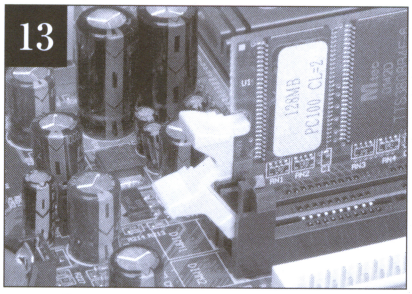
▲アップグレードと一緒にケースのクリーニングも済ませば，新品同様のマシンでWindows XPが楽しめるぞ!!



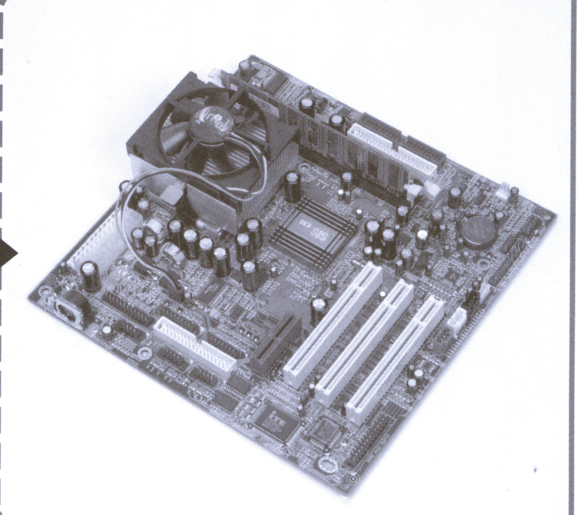
▲付属のテンションレバーで冷却フィンをCPUソケット両側のノブに引っかけに固定する



▲クーリングファンの電源ケーブルをマザー上のファン電源端子に接続



▲モジュールの両サイドにロックレバーで固定されていればOKだ



マザーボードをＰＣケースに組み込んだ後では，マザー上の各パーツチェックは，極めて困難だ。そのため，トラブルに発生源になりがちなＣＰＵ，メモリ，冷却フィン，クーリングファンなどの装着は，マザーのセットアップ時に必ず確認しておこう。「あわてず，急がず確実に!!」が基本だ

3 旧型モデルへのアップグレード・パーツの装着

今回の旧モデルのアップグレードでの最大の難関が，マザーボードの交換です。先ほど説明したように市販パソコンでは，独自仕様のマザーを搭載しているために，取り付けネジの位置がＡＴやＡＴＸフォームファクタなどの市販マザーとは異なります。このため，マザーボードの交換には，ボードの取り付けを工夫することになります。

交換マザーボードのＰＣケース設置方法

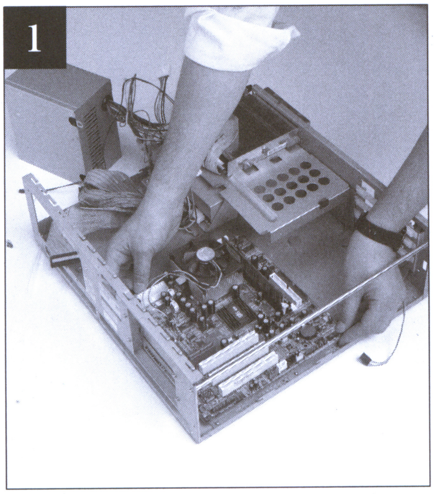
今回，アップグレードの対象にした富士通FMV機（デスクトップＰＣ）も例外ではなく，アップグレード・キットの「V630E」マザーへの換装には，ケースへ固定用のネジ穴を新たに開けるなどの下準備が必要です。

なお，「V630E」マザーは，主な端子類はすべてケーブルを介しリアパネルへ取り付けることが可能で，今回はＰＣＩ拡張スロットを利用しないので，電源ユニットやCD-Rドライブなどのフロントベイがじゃまにならないように注意すれば，基本的にはレイアウトは自由です。

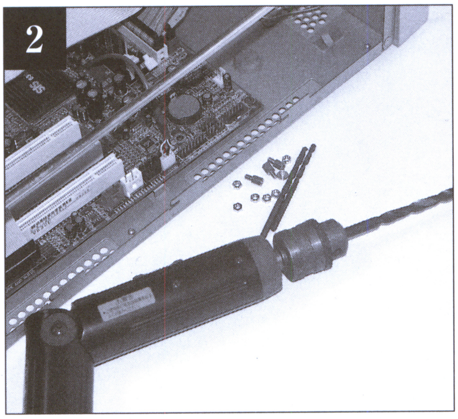
FMVデスクトップPCは，内部スペースが広く，マザーボードの換装には最適といえます。省スペースタイプの市販ＰＣでは，内部スペースが狭くマザーボードの換装に不向きな場合もありますので，アップグレードを行う場合には，マザーのサイズを調べ，設置スペースが確保できるか，事前に確認しておいたほうがよいでしょう。

ケースへのマザー取り付け穴の追加ですが，ＰＣケースは鉄板製で木材加工などに用いる手動ドリルでは手こずるようです。本格的な電動ドリルなどがあればよいのですが，手に入らなければ手動ド

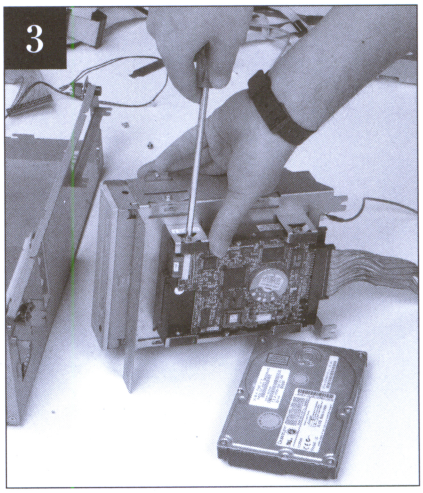
組み立て手順を忘れずに注意深くパーツを取り外そう!! 旧モデル・パソコンの分解



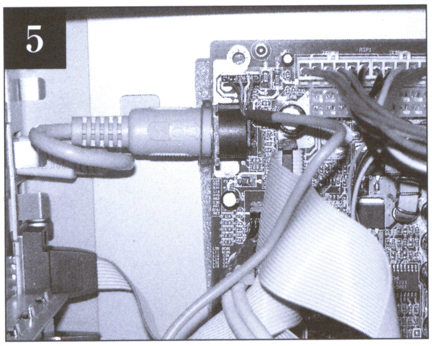
▲マザーボードの位置あわせを行う。Baby-ATマザーなので，大型きょう体の旧型FMV機には余裕だ



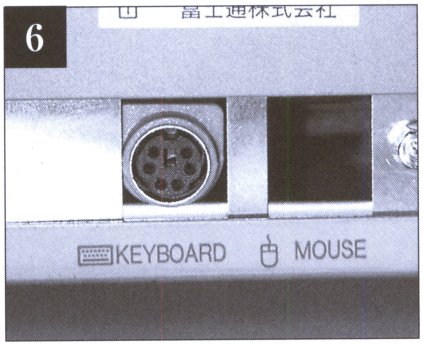
▲マザーの取り付け穴が合わない場合は，ドリルなどで穴開け作業が必要。トルクのある電動ドリルと鋼板ドリル歯があれば，簡単に行える。マザーボードの設置は，絶縁タイプのスペーサーを介して行おう



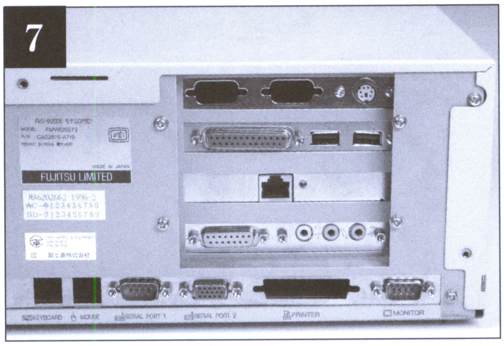
▲FMV内蔵のHDDを取り外し，新規に用意した大容量HDDに交換。Ultra ATA33/66対応ケーブルに付け替える



▲PS/2キーボードをつなぐために，ボード上のDIN端子をPS/2端子に変更するPS/2変換アダプタを利用した



▲変換アダプタのPS/2端子は，リアパネルの空き端子穴を利用し，固定しておけば，キーボードの接続が簡単になる



▲最新PCにも劣らない豊富なインターフェース端子が利用できるのも，本キットの特徴。ブロードバンド接続に使えるＬＡＮ端子もある

リルでも時間をかければ，穴開け加工は可能です。ただし，ドリル歯は必ず鋼板用に交換してください。

「V630E」のネジ穴位置は，Baby-ATフォームファクタで決められているものの，多少異なる場合も見られるようです。そのため，ＰＣケースにマザーを置いて取り付け穴の位置をマークし，それから穴を開ける，現物あわせで行えば間違いはないでしょう。なお，穴のサイズは3mm（直径）にしておきます。

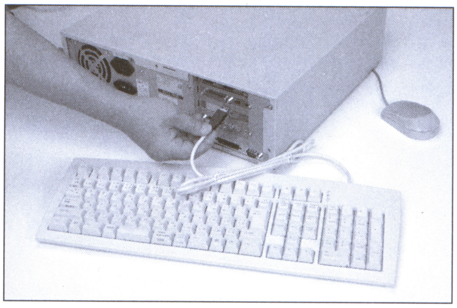
マザー取り付け時は，電子パーツショップなどにある絶縁タイプのスペーサーで固定します。金属タイプのスペーサーは，ケースとマザー間で不要なショートが起こる危険性もあるので，あまりお勧めできません。

HDD，CD-ROMドライブの取り付け&接続方法

数100MBほどしかないFMV内蔵ＨＤＤは，Windows XP導入には容量不足で，新たに用意した40GBタイプと交換します。前ページで分解したフロントドライブベイに取り付けるわけですが，IDEケーブルもマザーボード付属の新しいケーブルに交換しましょう。

FMVのＨＤＤ接続規格はE-IDEで，最近のUltra ATA（Ultra DMA）とは比較にならないほど低速です。換装した「V630E」マザーのIDE規格は，Ultra DMA 33/66となっており，最大66MB／秒（規格上）での高速なデータ転送が可能ですが，高速性能を生かすためには，この規格に対応した良質なIDEケーブルが不可欠です。

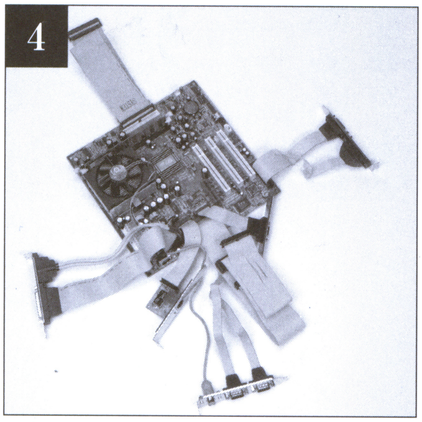
なお，FMV内蔵のＨＤＤも「V630E」マザーから使えますので，ＰＣケース内に取り付けスペースが確保できれば，データファイルのバックアップ用の低速ドライブとして生かすことも可能です。CD-ROMドライブは，IDEケーブルで接続しそのまま流用します。



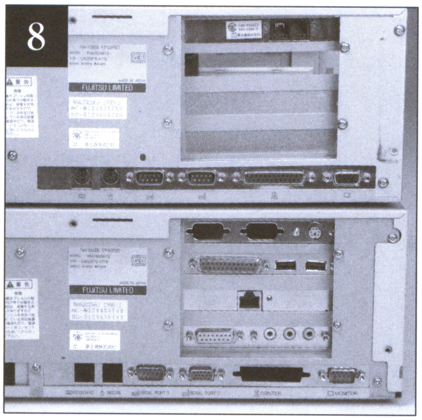
▲USB(1.1)端子もあり，キーボードやマウスのほかUSB対応外部ドライブなども増設できる

端子ケーブルを使って既存の端子穴に取り付ける

パラレル，シリアル，USB，ビデオ出力，サウンド入出力などのインターフェースは，マザーボード上の接続端子から専用ケーブルを介し，リアパネルに各端子を引き出すわけですが，「V630E」マザーは，拡張ボードのスロットカバーを利用し端子を増設する方式となっています。しかし，今回のFMV機の四つのスロットでは足らず，シリアル（2個）とビデオ出力は，これまで使っていたリアパネル下側の端子穴も流用し取り付けてあります。空いている端子穴が気になる場合は，シールなどでふさいでおけ



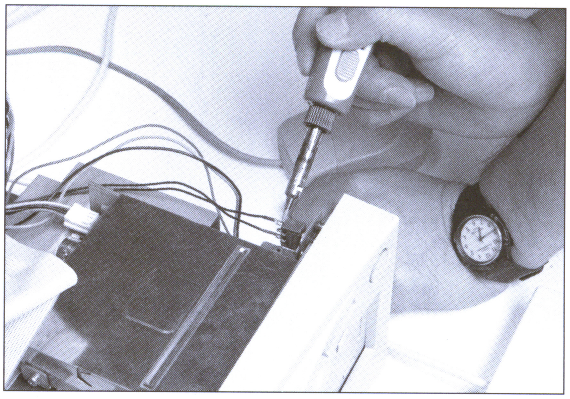
▲主なインターフェース端子は，マザーボード上から専用ケーブルでリアパネルに取り付ける



▲オリジナルFMV機（上）とアップグレード後の同機。以前はなかったUSBインターフェースも使えるようになった

FMV必須!!パワースイッチ交換と配線の変更

今回のFMV機では，フロントパネルのパワースイッチと電源ユニットが連動しないトラブルがあった。独自のオン／オフ回路を採用しているためで，スイッチを押しただけでは，電源が入らないのだ。そこで，オン／オフ・スイッチ（オルタネート型）スイッチに交換し，白・緑・黒の3本の配線のうち，緑と黒をショート／オープンすることで，電源ユニットを強制的に連動するようにした。なお，電源ユニットがＡＴ仕様のため，自動パワーオフには対応できない。



▶旧型FMV機の場合，パワースイッチの交換と配線の変更が必要になる

ば，防塵対策にもなります。

もう一つ注意したいのが，キーボード接続端子です。「V630E」などのATマザーは，この端子がDINタイプでしかもマザー上にあり，リアパネルに引き出せません。そこで，PCショップで売られているPS/2変換アダプタを介し，リアパネルにキーボード端子を取り付けてみました。このアダプタは，DIN端子を最近のキーボードで採用されているPS/2端子に変換する便利な小物パーツです。

Windows XPなどの最新ＯＳでは，スタートメニュー・キーなどと連動し，キーボードからスピーディな操作が行えるようになっています。

これらのキーを活用するにはPS/2規格のキーボードが必要で，PS/2変換アダプタを利用することでアップグレード機でも新型ＰＣ同様のキーボード操作を手に入れることができます。もちろん，USBキーボードも接続可能ですが，ＯＳインストール時はPS/2キーボードしか受け付けてくれず，この変換アダプタとPS/2キーボードの利用をお勧めします。

絶大なCPUパワーでWinXPの快適環境が実現

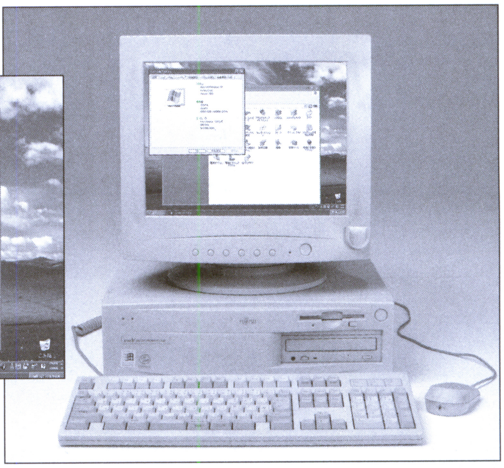
アップグレード終了後のFMV機にWindows XPを早速インストールしてみました。ひととおりの操作を行ってみた感想は，「予想を上回る軽快な動作」が体験できました。ベンチマークテストでも，ＣＰＵ，メモリ性能ともセレロン１ＧＨｚのパワーをまんべんなく引き出している結果となりました。

アップグレード・キットの「V630E」マザーは，ベースクロックやＣＰＵ倍率，ＰＣＩクロックなどの変更が，起動時に呼び出せるBIOS設定で行えるため，さらに高速なマシンへのチューンナップも行えるのも本キットのうれしい特徴といえるでしょう。

古くて使えず押入にしまったパソコンが，最新モデルにも劣らない高性能マシンとして復活できるこのアップグレード・キットは，まさにエコロジー時代のお助けツールといっても構わないでしょう。「羊の皮をかぶったハイパーマシンを!!」という人にもお勧めできます。

Win XPがサクサク動作する絶大なCPUパワー!!

仕上がったFMVアップグレード機の性能チェックをお馴染みのベンチマークソフト「HDBENCH」（EP82改／かず氏作）でテストしてみた。ALL／9779，Integer／45454，Float／45256のCPUパフォーマンスは，さすがにセレロン１ＧＨｚと思わせる結果。メモリ速度性能とともに，「V630E」マザーとの相性の良さが特徴的だ。この結果のとおり，これまでのＯＳでは最重力級のWindows XPも，軽快に操作することができた。メモリ容量をもっと増やせば，さらに快適感が増すはずだ。



▲重力級のWindows XPもCPUパワーで軽快な操作が体験できる

眠っているパソコンをWindows XPバージョンにしませんか?

旧OSパソコン
Windows3.1/95
CPU 386/486
ぐらいのパソコンを

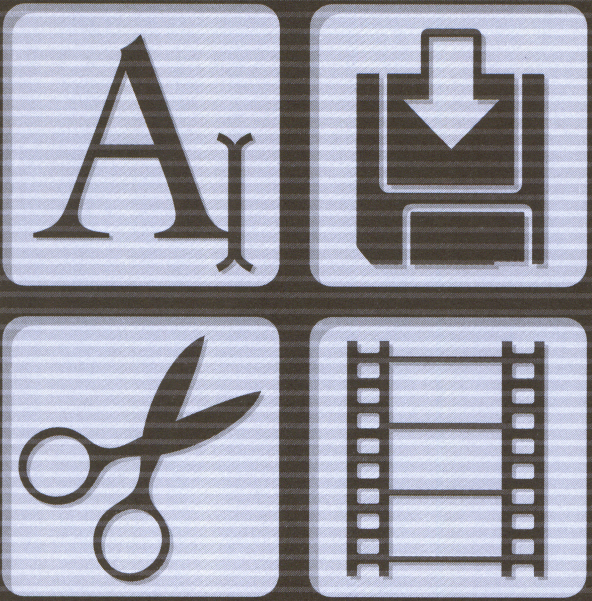
Grade Up
ハードだけなら3万円台位
ソフト込みでも4万円台位で
アップグレードできます。
（HDD、メモリー、CPU別途必要です。）

商　品
■ Windows XP Home(OEM)+V630Eマザーボード
=¥21,780（税送料別）
■ Windows XP Pro(OEM)+V630Eマザーボード
=¥27,780（税送料別）

V630E マザーボード	●サポートCPU：Intel Pentium Ⅲ FCPGA, Celeron PPGA or cyrix Ⅲ(Socket370)。Celeron 1GHzテスト済。●LAN：オンボード。●ビデオ：オンボード。●サウンド：オンボード。●電源：AT & ATX電源。●サイズ：Baby AT, 200mm×220mm。

お求めは eマート http://www.wakamatsu-net.com/

株式会社 若松通商
お問合せ先は、mail：sales@wakamatsu-net.com、又はTEL：03-3526-3677へ
※広告の価格には若干の変動がございます。ご了承ください。※すべての商品に消費税が加算されます。



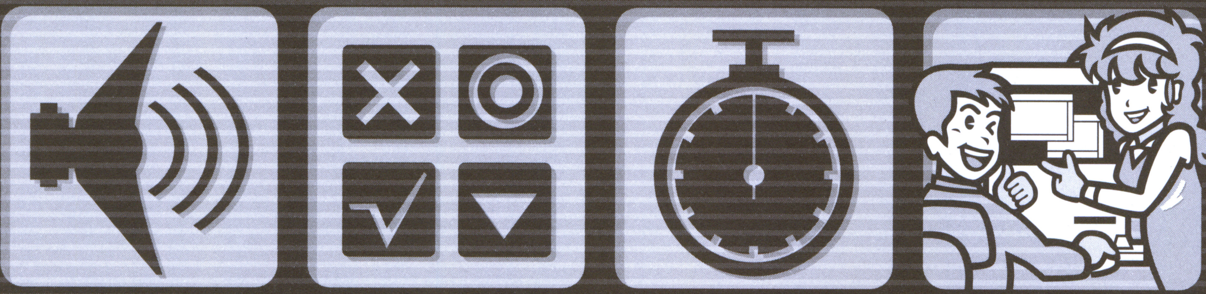
本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。

- Visual Basic
- Visual Basic+DirectX
- Visual C++
- C++Builder4
- F-BASIC V6.3
- HSP
- iアプリ

BM特選 投稿プログラムコーナー

プログラマー＆クリエイターへの登竜門



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2〜3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

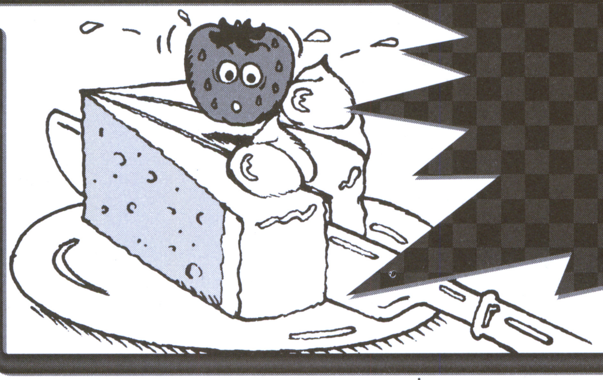
★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15 電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

掲載プログラムのダウンロード／投稿はこちら

http://www.basicmagazine.net/

Windows(Visual Basic V5.0)用



正しく切り分けよう Cut!（カット）

木村　雄一（20歳）

ストーリー

ある所に，普段はとても仲の良い姉弟がおりました。しかし,おやつの時間になると，いつもケンカばかりしています。
「ああっ，お姉ちゃんのケーキのほうが大きい！！」
「何言ってるの！ 二つとも同じ大きさでしょうが！！」

そんな状況を見かねた母は，おやつを均等に切る訓練をはじめたのでありました…。

ルール説明

画面におやつの形が現れるので，それを指定された個数に分割します。

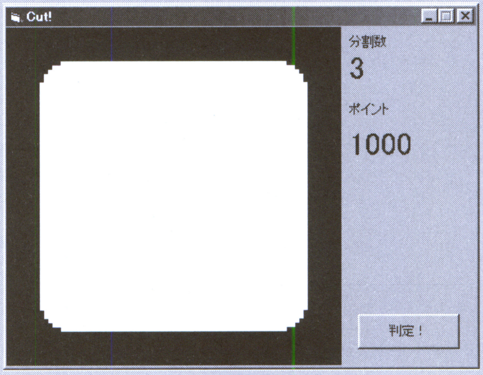
分割したおやつの中で最大と最小の面積の差が，平均面積の10%以内なら合格になります。また，5%以内に収まると特別ボーナスとして500ポイント与えられます。

もし合格ラインを超えた場合は，「差の面積ー合格ライン」が引かれ，もう一度やり直しとなります。もし分割数をまちがえた場合や，ポイントが0を下回った場合はゲーム・オーバーです。

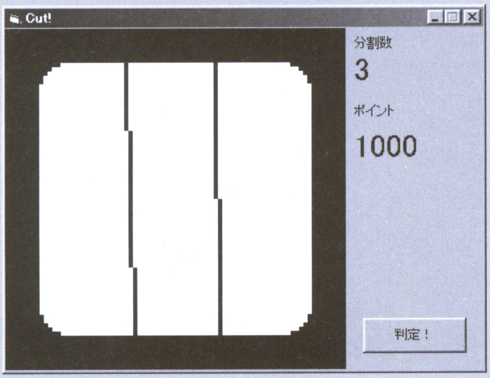
操作方法

おやつの画面が出たら，まず切る始点を決めます。画面をクリックすると赤い

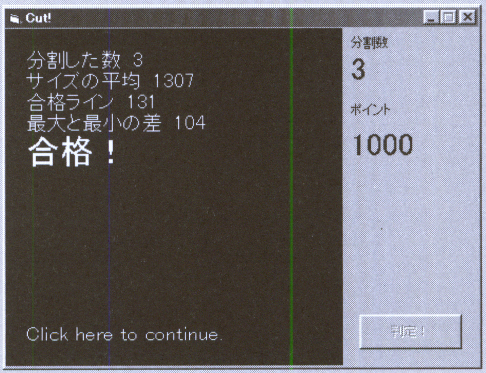
《写真1》おやつをCutしていくゲーム



《写真2》最初は、この形のおやつを3分割する



《写真3》こんな感じで切りわける。果たして結果は…



《写真4》見事合格！この調子で進めていこう

《OVER FLOW》一段と寒さも厳しくなる時期，みなさんはいかがお過ごしでしょうか。寒い日は，暖かい部屋でベーマガ（マイコンBASICマガジンの略称）を読む，プログラミングに熱中するなんてのはどう？今月も，寒さを吹き飛ばしてしまう内容で，お届けいたします。

点が打たれます(これが始点の位置)。つぎにクリックすると，始点からクリックした場所まで切り込みが入ります。こうして指定された個数に分割してください。

判定アルゴリズムの関係上，1ドットでもつながっている場合は，一つのおやつとして扱われます。切るときは注意してください。

指定した個数に切れたら，画面右の「判定！」をクリックしてください。前述のルールにしたがって得点が計算されます。

入力方法

フォームにはコマンドボタン，ラベル，ピクチャボックスを配置します。おやつのデータは80×80,8ビットBMPを使います。

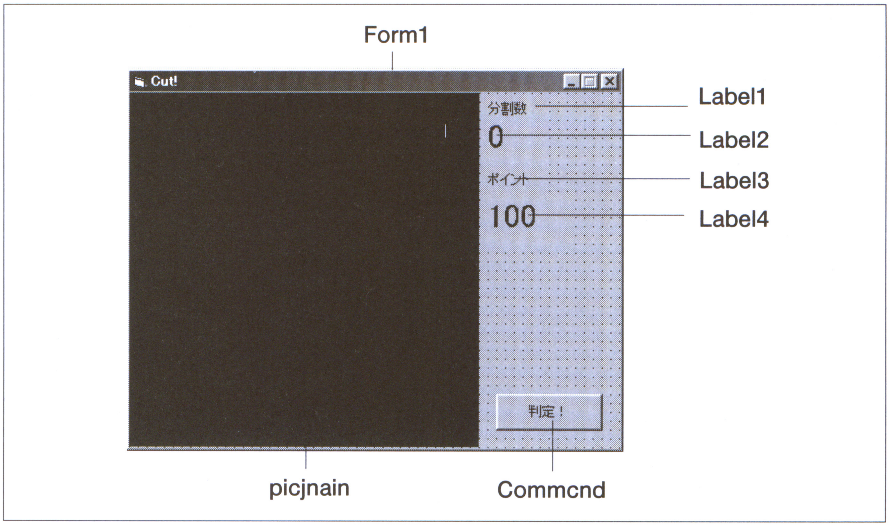
何もない部分はパレット番号「0」，おやつ本体はそれ以外のパレット番号で描いてください。

プログラムについて

「判定が厳しい！」と思う人は，変数bdvalの値を適当に変えてください。ただ，あんまり簡単過ぎるとゲームのバランスが崩れます。また，各ステージの分割数は，cut_count配列に入っています。

面積の算出には幅優先探索を使っています。ちょっと効率が悪いかもしれませんが，それほど大きな面積を扱うわけではないので問題はないと思います。

おやつをカットする部分は，マイコンBASIC MAGAZINE2001年10月号「チャレンジ！！ワンダーウィッチ」のライン描画アルゴリズムを参考にさせていただきました。



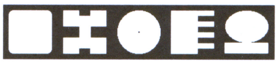
《第1図》フォームの配置図

最後に

均等に切れたと思っても，結構バラバラなもんですね(笑)。おやつのデータはビットマップなので，簡単にお絵描き感覚で作れます。ぜひいろいろなデータを作って試してください。

★参考にしたもの

・マイコンBASIC MAGAZINE2001年10月号「チャレンジ！！ワンダーウィッチ」(松原拓也)



左からSTG1.bmp～STG5.bmpの画像

《第2図》サンプル画像のビットマップ

CHECKER FLAG

編：物体を切りわけるというゲーム・アイデアがいいですね。個性のあるゲームです。

つぐ美：ただ，操作性の悪さも目に付いちゃいますね。このゲームの肝になる切り分けの操作が，やはりイマイチ。もうひと工夫ほしいところです。また，合格ラインがチョット厳しいので，なかなか先へ進まないのも難点ね。

影：切りわけた一つ一つのおやつのサイズが，表示されるといいんじゃないかな？

Dr.D：この作品も，テストプレイが不足している印象じゃ。ゲーム作品の場合は，ある程度形にしたあとのバランス調整が大事じゃゾ。

《第１表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	Cut!
Height	4815
Left	45
LinkTopic	Form1
MaxButton	0 'False
ScaleHeight	321
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	449
StartUpPosition	3 'Windows の既定値
Top	330
Width	6735
コマンドボタンの設定	
Name	Command1
Caption	判定！
Height	33
Left	336
TabIndex	1
Top	272
Width	97
ピクチャボックスの設定	
Name	picMain
AutoRedraw	-1 'True
BackColor	&H00000000&
BorderStyle	0 'なし
FillStyle	0 '塗りつぶし
Height	320
Left	0
ScaleHeight	320
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	320
TabIndex	0
Top	0
Width	320
ラベルの設定	
Name	Label1
Caption	分割数
Height	17
Left	328
TabIndex	2
Top	8
Width	49
Name	Label2
Caption	0
Font	ＭＳ Ｐゴシック
…Size	22
Height	41
Left	328
TabIndex	3
Top	24
Width	49
Name	Label3
Caption	ポイント
Height	17
Left	328
TabIndex	4
Top	72
Width	49
Name	Label4
Caption	100
Font	ＭＳ Ｐゴシック
…Size	22
Height	41
Left	328
TabIndex	5
Top	96
Width	73

《リスト1》「Cut」のフォームのプログラム・リスト

```
Const GAME_TITLE = 1
Const GAME_CLK1 = 2
Const GAME_CLK2 = 3
Const GAME_RESULT = 4
Const GAME_CLEAR = 5
Dim x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer
Dim game_sw As Integer, nstg As Integer, gflag As Integer, npoint As Integer
Sub draw_map()
    '現在読み込まれているマップの表示
    Dim ix As Integer, iy As Integer
    picMain.Cls
    For iy = 0 To 79
        For ix = 0 To 79
            If map(ix + 1, iy + 1) = 1 Then
                picMain.Line (ix * 4, iy * 4)-(ix * 4 + 3, iy * 4 + 3), vbWhite, BF
            End If
        Next
    Next
End Sub

Private Sub Command1_Click()
    Dim avg As Integer, sum As Integer, i As Integer
    Dim max As Integer, min As Integer
    Dim bdval As Integer
    '判定
    judge
    '結果の表示
    picMain.Cls
    picMain.FontSize = 14
    print_str "分割した数 " + str(j_count), 20, 20, vbWhite
    If j_count <> cut_count(nstg) Then
        '分割数が違う場合
        print_str "分割数が違います!!", 20, 40, vbRed
        picMain.FontSize = 24
        print_str "GAME OVER", 20, 60, vbRed
        gflag = 1
    Else
        '分割数が合っている場合
        sum = 0
        max = 0
        min = 9999
        For i = 0 To j_count - 1
            If max < j_area(i) Then max = j_area(i)
            If min > j_area(i) Then min = j_area(i)
            sum = sum + j_area(i)
        Next
        avg = sum / j_count
        bdval = avg * 0.1
        print_str "サイズの平均 " + str(avg), 20, 40, vbWhite
        print_str "合格ライン " + str(bdval), 20, 60, vbWhite
        print_str "最大と最小の差 " + str(max - min), 20, 80, vbWhite
        picMain.FontSize = 24
        If max - min > bdval Then
            print_str "減点!" + str(bdval - (max - min)) + "point", 20, 100, vbRed
            npoint = npoint + (bdval - (max - min))
            If npoint < 0 Then
                print_str "GAME OVER", 20, 140, vbRed
                gflag = 1
            Else
                gflag = 2
            End If
        ElseIf max - min < avg * 0.05 Then
            print_str "Bonus 500point!!!", 20, 100, vbYellow
            npoint = npoint + 500
            gflag = 0
        Else
            print_str "合格!", 20, 100, vbYellow
            gflag = 0
        End If
    End If
    picMain.FontSize = 14
    print_str "Click here to continue.", 20, 280, vbWhite
    game_sw = GAME_RESULT
    Command1.Enabled = False
End Sub

Private Sub Form_Load()
    Command1.Enabled = False
    game_sw = GAME_TITLE
    nstg = 0
    npoint = 1000
    cut_count(0) = 3
    cut_count(1) = 3
    cut_count(2) = 5
    cut_count(3) = 5
    cut_count(4) = 7
    Label4.Caption = npoint
    show_title
End Sub
Sub show_title()
    'タイトル画面の表示
    picMain.Cls

    picMain.FontSize = 48
    print_str "C u t !", 60, 50, vbWhite
    picMain.FontSize = 16
    print_str "Click here to start.", 70, 180, vbWhite

    picMain.FontSize = 12
    print_str "- Presented by Yuichi Kimura 2001 -", 30, 250, vbWhite

End Sub
Sub print_str(str As String, ByVal x As Integer, ByVal y As Integer, ByVal col As Long)
    picMain.ForeColor = col
    picMain.CurrentX = x
    picMain.CurrentY = y
    picMain.Print str
End Sub

Private Sub picMain_MouseDown(Button As Integer, Shift As Integer, x As Single, y As
Single)
    Select Case game_sw
        Case GAME_TITLE:
            'ゲームの開始
            npoint = 1000
            nstg = 0
            stage_init
            game_sw = GAME_CLK1
        Case GAME_CLK1:
```

★初投稿です，どうも。んー，プログラムの基本すら知らないのにベーマガです。複雑です。まねごとならできるけど。投稿って恥ずかしい。（福岡県山門郡・**瞬間移動14歳**）……【**つぐ美**：だからベーマガなんでしょ。がんばって勉強して投稿してね。よろしく！】

```
            x1 = Int(x / 4)
            y1 = Int(y / 4)
            If x1 < 0 Then x1 = 0
            If y1 < 0 Then y1 = 0
            '印を付ける
            picMain.ForeColor = vbRed
            picMain.FillColor = vbRed
            picMain.Circle (x1 * 4 + 2, y1 * 4 + 2), 4
            game_sw = GAME_CLK2
        Case GAME_CLK2:
            x2 = Int(x / 4)
            y2 = Int(y / 4)
            If x2 < 0 Then x2 = 0
            If y2 < 0 Then y2 = 0
            cut_map x1, y1, x2, y2
            draw_map
            game_sw = GAME_CLK1
        Case GAME_RESULT:
            If gflag = 1 Then
                'ゲームオーバー
                show_title
                game_sw = GAME_TITLE
            ElseIf gflag = 2 Then
                '同じステージのやり直し
                stage_init
                game_sw = GAME_CLK1
            Else
                'クリア
                nstg = nstg + 1
                If nstg = 5 Then
                    'おめでとうのメッセージ
                    picMain.Cls
                    Label4.Caption = npoint
                    picMain.FontSize = 24
                    print_str "Congratulations!", 40, 0, vbYellow
                    picMain.FontSize = 14
                    print_str "Your point:" + str(npoint), 20, 40, vbWhite
                    print_str "Click here to continue.", 20, 280, vbWhite
                    game_sw = GAME_CLEAR
                Else
                    stage_init
                    game_sw = GAME_CLK1
                End If
            End If
        Case GAME_CLEAR:
            show_title
            game_sw = GAME_TITLE
    End Select
End Sub
Sub stage_init()
    If load_map(nstg) = False Then
        MsgBox "データの読み込みに失敗しました。"
        End
    End If
    draw_map
    Label2.Caption = cut_count(nstg)
    Command1.Enabled = True
    Label4.Caption = npoint
End Sub
```

《リスト2》「Cut」のモジュールのプログラム・リスト

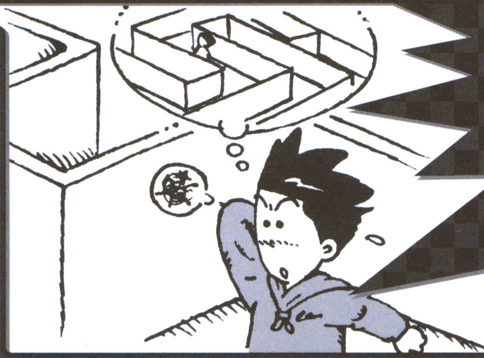
```
Type BITMAPFILEHEADER
        bfType As Integer
        bfSize As Long
        bfReserved1 As Integer
        bfReserved2 As Integer
        bfOffBits As Long
End Type
Type BITMAPINFOHEADER
        biSize As Long
        biWidth As Long
        biHeight As Long
        biPlanes As Integer
        biBitCount As Integer
        biCompression As Long
        biSizeImage As Long
        biXPelsPerMeter As Long
        biYPelsPerMeter As Long
        biClrUsed As Long
        biClrImportant As Long
End Type
Type POINTAPI
        x As Long
        y As Long
End Type
Dim queue(1000) As POINTAPI, qhead As Integer, qtail As Integer
Public j_count As Integer, j_area(10) As Integer
Public cut_count(5) As Integer
Public map(82, 82) As Byte, map_flag(82, 82) As Byte
Function load_map(ByVal stg As Integer) As Boolean
    Dim fname(5) As String, bfh As BITMAPFILEHEADER, bih As BITMAPINFOHEADER
    Dim err As Byte, offset As Long, ix As Integer, iy As Integer
    Dim bmpdata(0 To 80 * 80 - 1) As Byte
    fname(0) = "stg1.bmp"
    fname(1) = "stg2.bmp"
    fname(2) = "stg3.bmp"
    fname(3) = "stg4.bmp"
    fname(4) = "stg5.bmp"
    err = 0
    Open fname(stg) For Binary As #1
        Get #1, , bfh
        Get #1, , bih
        If bih.biWidth = 80 And bih.biHeight = 80 Then
            offset = bfh.bfOffBits + 1
            Get #1, offset, bmpdata
        Else
            err = 1
        End If
    Close #1
    If err = 1 Then
        load_map = False
    Else
        'マップ配列に移す
        For iy = 0 To 79
            For ix = 0 To 79
                If bmpdata((79 - iy) * 80 + ix) = 0 Then
```


★1月号のCPUの歴史はすごく役立ちました。なんか「別冊」にして，もっとくわしくしてもいいくらい。こういうことって，パソコンを作るうえで大切ですよね。(神奈川県川崎市・**桜井道孝21歳**)……**【編**：この企画，大好評でした。まだ読んでいない人はぜひ！**】**

```
                map(ix + 1, iy + 1) = 0
            Else
                map(ix + 1, iy + 1) = 1
            End If
        Next
    Next
    load_map = True
  End If
End Function

Sub cut_map(ByVal x1 As Integer, ByVal y1 As Integer, ByVal x2 As Integer, ByVal y2 As Integer)
    Dim dx As Integer, dy As Integer, tmp As Integer
    Dim ix As Integer, nx As Integer, ny As Integer, iy As Integer
    Dim sx As Integer, sy As Integer
    Dim flg As Integer
    dx = Abs(x2 - x1)
    dy = Abs(y2 - y1)
    If x2 - x1 > 0 Then sx = 1 Else sx = -1
    If y2 - y1 > 0 Then sy = 1 Else sy = -1
    flg = 0
    ny = y1
    nx = x1
    If dy < dx Then
        While nx <> x2
            map(nx + 1, ny + 1) = 0
            nx = nx + sx
            flg = flg + 2 * dy
            If flg >= dx Then
                ny = ny + sy
                flg = flg - dx * 2
            End If
        Wend
    Else
        While ny <> y2
            map(nx + 1, ny + 1) = 0
            ny = ny + sy
            flg = flg + 2 * dx
            If flg >= dy Then
                nx = nx + sx
                flg = flg - dy * 2
            End If
        Wend

    End If
End Sub
Sub judge()
    '分割の判定
    Dim ix As Integer, iy As Integer, i As Integer
    Dim area As Long
    Dim sx(4) As Integer, sy(4) As Integer
    Dim np As POINTAPI, np2 As POINTAPI

    sx(0) = 0: sy(0) = -1
    sx(1) = 0: sy(1) = 1
    sx(2) = 1: sy(2) = 0
    sx(3) = -1: sy(3) = 0

    qhead = 0: qtail = 0
    j_count = 0
    For iy = 0 To 81
        For ix = 0 To 81
            map_flag(ix, iy) = 0
        Next
    Next
    For iy = 0 To 79
        For ix = 0 To 79
            If map(ix + 1, iy + 1) = 1 And map_flag(ix + 1, iy + 1) = 0 Then
                '面積を求める
                area = 0
                np.x = ix + 1
                np.y = iy + 1
                map_flag(ix + 1, iy + 1) = 1
                enqueue np
                Do While isempty = False
                    np = dequeue
                    area = area + 1
                    For i = 0 To 3
                        np2.x = np.x + sx(i)
                        np2.y = np.y + sy(i)
                        If map(np2.x, np2.y) = 1 And map_flag(np2.x, np2.y) = 0 Then
                            map_flag(np2.x, np2.y) = 1
                            enqueue np2
                        End If
                    Next
                    DoEvents
                Loop
                If j_count < 10 Then
                    j_area(j_count) = area
                End If
                j_count = j_count + 1
            End If
        Next
    Next
End Sub
Sub enqueue(item As POINTAPI)
    queue(qtail) = item
    qtail = qtail + 1
    If qtail = 1000 Then qtail = 0
End Sub
Function dequeue() As POINTAPI
    dequeue = queue(qhead)
    qhead = qhead + 1
    If qhead = 1000 Then qhead = 0
End Function
Function isempty() As Boolean
    If qhead = qtail Then
        isempty = True
    Else
        isempty = False
    End If
End Function
```

★買っちゃいました。「ピクミン」のCMソングCD。（茨城県古河市・**年齢不相応41歳**）……**【Dr.D：**「愛のうた〜ピクミンCMソング」ってやつじゃな。♪ぼくたちピクミン，あなただけについていく〜。**影：**どひゃー！音痴！でも，なぜDr.が？！**Dr.D：**ウシシ。**】**

Windows(Visual Basic V6.0)用



懐かしいタイプの3D迷路

迷路（めいろ）

戸田　文彰（22歳）

内容

いたってシンプルな擬似3Dの迷路です。シンプルですが，意外と方向感覚を失います。

遊びかた

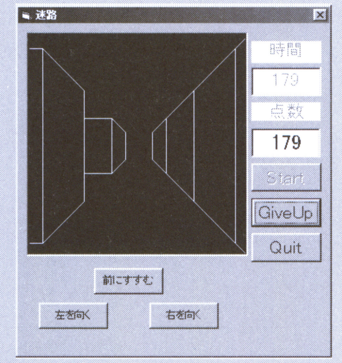
ゲームを起動したら画面右上にある「Start」ボタンを押してください。

ゲームが始まったら，画面下のほうにある「前にすすむ」「右を向く」「左を向く」ボタンでゴールへ向かってください。

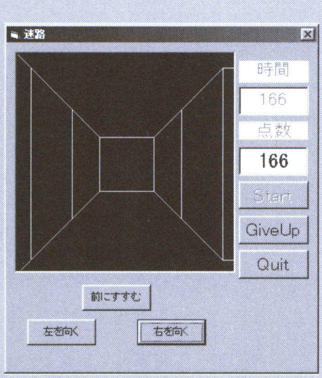
制限時間があります。基本的には3分です。どうしても時間が足りない場合は，「Form_Load」にある“Time”の値を増やしてください。

ゴールは黄色の四角が表示されます。

《写真1》VBで昔懐かしの迷路を再現した



《写真2》制限時間内にゴールを目指そう



《写真3》情報が少ないため，かなり迷いやすい

《写真4》難易度は高め。影さんはギブアップ

操作方法

「前にすすむ」は，前にすすみます。「右を向く」は右に，「左を向く」は左に向きを変えます。

したがって左や右に曲がりたい場合は，一度向きを変えてから前にすすむようにしてください。

「GiveUp」ボタンは，途中でやめるときに使用してください。「Quit」で終了します。

改造のアドバイス

迷路のマップ・ファイルは，外部にあるMapdat.dat(リスト2)を読み込み，表

CHECKER FLAG

影：昔ながらの3D迷路。画面表示などはしっかり作られているけど，やはりゲームとしては，もの足りないなぁ。

つぐ美：フォーム上のボタンで操作をするので，操作性もイマイチね。また，地図表示や向いている方向表示などが全然ないため，クリアするのはチョット むずかしいわ。

Dr.D:リストはしっかりと記述されているほうなので，問題はやはりゲーム性じゃな。他の人に遊んでもらったり，ある程度日にちをおいて自分で再度テストプレイをして，改良を重ねていくようにしよう。

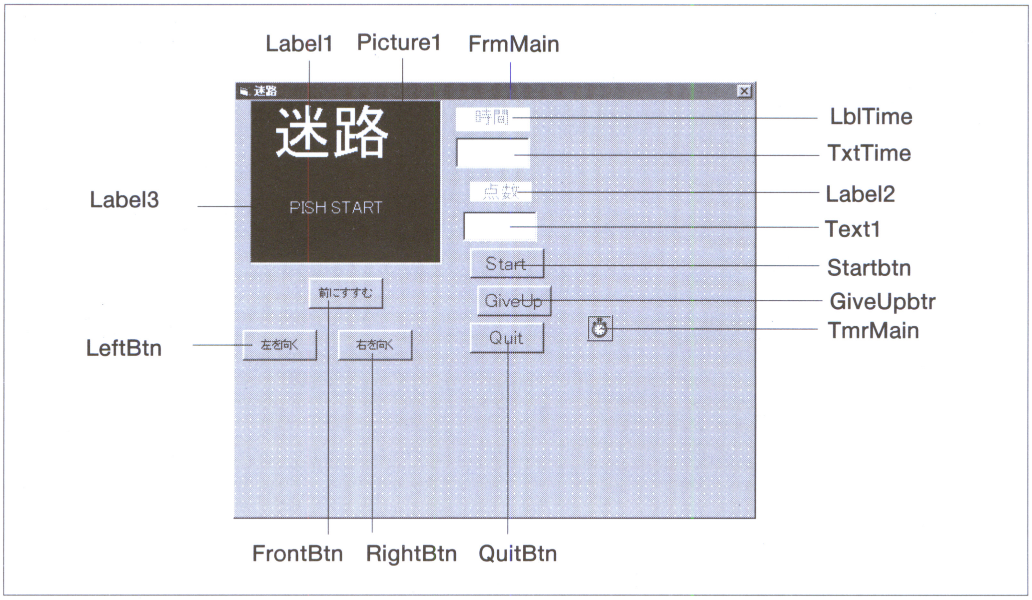
編：一応プロパティ表を掲載してはいますが，このプログラムは一部のプロパティをリスト上から設定しますので，その部分はプロパティを設定する必要はありません。

★期末試験も終わったし，ゲーム作り再開！と思ったら，もうすぐ受験だー。（京都市・Essa14歳）……【編：高校受験ですね。風邪などひかずに体調を十分管理してがんばってください。影：それを乗り越えると，また3年後には大学受験がぁぁ。編：ドタっ！】

示しています。ですので，自分で好きなマップを読み込むようにしたり，自動迷路作成機能を搭載してみたりなど，各自で改造してみてはいかがでしょうか。

今，自分で付け足そうと思っているのが，上の二つの機能に加えて，ある一定の時間がたったら(残り1分とか)地図を表示する機能をつけようと思っています。まだ，思案中ですが…。

シンプルなのでいろいろと改造ができると思います。



《第1図》フォームの配置図

《第1表》プロパティ表

フォームの設定	
Name	frmMain
AutoRedraw	-1 'True
BorderStyle	1 '固定(実線)
Caption	迷路
Font	ＭＳ Ｐゴシック
…Size	22
ForeColor	&H00FFFFFF&
Height	8145
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	518
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	625
StartUpPosition	2 '画面の中央
Top	0
Width	9465
テキストボックスの設定	
Name	Text1
Alignment	2 '中央揃え
Font	ＭＳ Ｐゴシック
…Size	18
Height	32
Left	248
TabIndex	11
Top	120
Width	81
Name	TxtTime
Alignment	2 '中央揃え
Font	ＭＳ Ｐゴシック
…Size	16
ForeColor	&H00FF8080&
Height	33
Left	240
TabIndex	4
Top	40
Width	81
コマンドボタンの設定	
Name	Quitbtn
Caption	Quit
Font	ＭＳ Ｐゴシック
…Size	16
Height	33
Left	256
TabIndex	9
Top	240
Width	81
Name	GiveUpbtn
Caption	GiveUp
Enabled	0 'False
Font	ＭＳ Ｐゴシック
…Size	16
Height	33
Left	264
TabIndex	8

Top	200
Width	81
Name	Startbtn
Caption	Start
Font	ＭＳ Ｐゴシック
…Size	16
Height	32
Left	256
TabIndex	7
Top	160
Width	81
Name	RightBtn
Caption	右を向く
Enabled	0 'False
Font	ＭＳ Ｐゴシック
…Size	9
Height	33
Left	112
TabIndex	3
Top	248
Width	81
Name	LeftBtn
Caption	左を向く
Enabled	0 'False
Font	ＭＳ Ｐゴシック
…Size	9
Height	33
Left	8
TabIndex	2
Top	248
Width	81
Name	FrontBtn
Caption	前にすすむ
Enabled	0 'False
Font	ＭＳ Ｐゴシック
…Size	9
Height	33
Left	80
TabIndex	1
Top	192
Width	81
ピクチャボックスの設定	
Name	Picture1
AutoRedraw	-1 'True
BackColor	&H00000000&
Font	ＭＳ Ｐゴシック
…Size	9
Height	177
Left	16
ScaleHeight	173
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	205
TabIndex	0
Top	0
Width	209
ラベルの設定	
Name	Label1

Alignment	2 '中央揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	迷路
Font	ＭＳ Ｐゴシック
…Size	48
ForeColor	&H00FFFFFF&
Height	64
Left	22
TabIndex	6
Top	0
Width	129
Name	Label2
Alignment	2 '中央揃え
AutoSize	-1 'True
BackColor	&H00FFFFFF&
Caption	点数
Font	ＭＳ Ｐゴシック
…Size	16
ForeColor	&H00FF8080&
Height	21
Left	256
TabIndex	10
Top	88
Width	67
Name	Label3
Alignment	2 '中央揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	PISH START
Font	ＭＳ Ｐゴシック
…Size	14
ForeColor	&H00FFFFFF&
Height	19
Left	40
TabIndex	12
Top	104
Width	105
Name	LblTime
Alignment	2 '中央揃え
BackColor	&H00FFFFFF&
Caption	時間
Font	ＭＳ Ｐゴシック
…Size	14
ForeColor	&H00FF8080&
Height	25
Left	240
TabIndex	5
Top	8
Width	81
タイマーの設定	
Name	TmrMain
Enabled	0 'False
Interval	1000
Left	384
Top	232

★Windows XPのことをもっとくわしく載せてほしい。(新潟県長岡市・南　寿和19歳) ……【影：では，私にXPを買ってくれ。そこからはじめよう。編：どういうことですか，読者に向かって言うセリフじゃありませんよ。影：じゃ編くん，買ってくれるか？編：…。】

《リスト1》「迷路」のプログラム・リスト

```
Option Explicit

Dim Time As Long
Dim Score As Long
Dim mx As Long, my As Long  ' 自分の位置
Dim mh As Long                ' 自分の方向
Dim mp(4) As Long             ' ３次元描画用
Dim ms(19) As String           ' 迷路のデータ
Dim maze(19, 19) As Long      ' 迷路のデータ

Private Sub Form_Load()

    Call Parts_Set

'2重起動禁止
    If App.PrevInstance = True Then Unload Me

'時間と点数の初期設定
    Time = 180
    Score = 180

End Sub

Private Sub Parts_Set()

    Dim xx As Long, yy As Long   'フォーム(frmMain)のサイズ設定の為に使用。


'フォーム(frmMain)のサイズ設定(360x380)
    xx = Screen.TwipsPerPixelX
    yy = Screen.TwipsPerPixelY

    With frmMain
        .Width = frmMain.Width - (frmMain.ScaleWidth * xx) + 360 * xx
        .Height = frmMain.Height - (frmMain.ScaleHeight * yy) + 380 * yy
    End With

'その他もろもろの設定

    With TxtTime              '残り時間表示
        .Top = 50
        .Left = 270
        .Text = 180
    End With

    With LblTime
        .Top = 20
        .Left = 270
    End With

    With Picture1             '迷路等表示部分
        .Top = 10
        .Left = 10
        .Width = 256
        .Height = 256
    End With
```

```
With FrontBtn           '「前にすすむ」ボタン
    .Top = 280
    .Left = 88
    .Width = 80
    .Height = 30
End With

With LeftBtn            '「左を向く 」ボタン
    .Top = 320
    .Left = 24
    .Width = 80
    .Height = 30
End With

With RightBtn           '「右を向く」ボタン
    .Top = 320
    .Left = 152
    .Width = 80
    .Height = 30
End With

With Startbtn           '「Start」ボタン
    .Top = 160
    .Left = 270
End With

With GiveUpbtn          '「GiveUp」ボタン
    .Top = 200
    .Left = 270
End With

With Quitbtn            '「Quit」ボタン
    .Top = 240
    .Left = 270
End With

With Label2
    .Top = 90
    .Width = 80
    .Left = 270
End With

With Label1
    .Top = 50
    .Left = 40
End With

With Label3
    .Top = 180
    .Left = 75
End With

With Text1
    .Top = 120
    .Left = 270
    .Width = 80
    .Text = 180
End With
```

★最近，Linux→レポート作成，Windows→遊び（ゲーム）とわかれてきた。（秋田県横手市・**ハングンマル20歳**）……【**つぐ美**：2台持ってるの？それとも，デュアルOSってやつ？**影**：私の場合，Win98はゲーム，Win2000は投稿チェック，WinMeは秘密！】

```
End Sub
Private Sub Init()

'ゲームが始まった時

    Label1.Visible = False
    Label3.Visible = False

    FrontBtn.Enabled = True
    LeftBtn.Enabled = True
    RightBtn.Enabled = True
    Startbtn.Enabled = False

    Call First_Init

End Sub
Private Sub First_Init()

Dim i As Long, j As Long

'迷路の表示に使用する配列の初期化
mp(0) = 0
mp(1) = 16
mp(2) = 64
mp(3) = 96
mp(4) = 112

'迷路のデータ読み込み
Open "Mapdata.dat" For Input As #1 '" Mapdata.dat" は自分で作成することができます。
２０×２０

  For i = 0 To 19
    Input #1, ms(i)
  Next i

  Close #1
  For i = 0 To 19
    For j = 0 To 19
        If Mid(ms(j), i + 1, 1) = "1" Then
            maze(i, j) = 1
        ElseIf Mid(ms(j), i + 1, 1) = "2" Then
            maze(i, j) = 2
        Else
            maze(i, j) = 0
        End If
    Next j
  Next i


'あなたが最初にいる場所（座標）と向き
    mx = 1
    my = 18
    mh = 1

    Call Draw
```

```
End Sub
Private Sub Draw()

    Dim i As Long, j As Long

    Picture1.BackColor = &H0
    Picture1.Cls

  For i = 1 To 5
    j = i
    If Check(0, i) Then Exit For
  Next

  ' 左右の壁の縦線を描画
  For i = 1 To j
    If i < 5 Then
      Picture1.Line (mp(i), mp(i))-(mp(i), 255 - mp(i)), &HFFFFFF
      Picture1.Line (255 - mp(i), mp(i))-(255 - mp(i), 255 - mp(i)), &HFFFFFF
    End If
  Next

  ' 突き当たりの壁の上下の線を描画
  If j < 5 Then
    Picture1.Line (mp(j), mp(j))-(255 - mp(j), mp(j)), &HFFFFFF
    Picture1.Line (mp(j), 255 - mp(j))-(255 - mp(j), 255 - mp(j)), &HFFFFFF
    If Check(0, j) = 2 Then ' ゴールの壁には黄色の四角を描画
      i = 128 - Int((128 - mp(j)) * 0.7)
      Picture1.Line (i, i)-(255 - i, 255 - i), &HFFFF&, BF
    End If
  End If

  ' １つ横を調べて天井の線を壁か通路にする
  For i = 1 To j
    If i < 5 Then
      If Check(-1, i - 1) Then ' 右側壁
        Picture1.Line (mp(i), mp(i))-(mp(i - 1), mp(i - 1)), &HFFFFFF
        Picture1.Line (mp(i), 255 - mp(i))-(mp(i - 1), 255 - mp(i - 1)), &HFFFFFF
      Else                    ' 右側通路
        Picture1.Line (mp(i), mp(i))-(mp(i - 1), mp(i)), &HFFFFFF
        Picture1.Line (mp(i), 255 - mp(i))-(mp(i - 1), 255 - mp(i)), &HFFFFFF
      End If
      If Check(1, i - 1) Then  ' 左側壁
        Picture1.Line (255 - mp(i), mp(i))-(255 - mp(i - 1), mp(i - 1)), &HFFFFFF
        Picture1.Line (255 - mp(i), 255 - mp(i))-(255 - mp(i - 1), 255 - mp(i - 1)),
&HFFFFFF
      Else                    ' 左側通路
        Picture1.Line (255 - mp(i), mp(i))-(255 - mp(i - 1), mp(i)), &HFFFFFF
        Picture1.Line (255 - mp(i), 255 - mp(i))-(255 - mp(i - 1), 255 - mp(i)),
&HFFFFFF
      End If
    End If
  Next

End Sub
Function Check(ByVal x As Integer, ByVal y As Integer) As Integer

Select Case mh
    Case 0
```


★お初です。初心者です。プログラミングってどんなソフトでやるんですか。IEでHTMLやJava Scriptしかやってません。フリーウエアはダウンロードできないのを前提に教えてほしいです。（福島県二本松市・**加藤K子15歳**）……**【影：待て！数か月後の本誌！】**

```
        Check = maze(mx + x, my - y)
    Case 1
        Check = maze(mx + y, my + x)
    Case 2
        Check = maze(mx - x, my + y)
    Case 3
        Check = maze(mx - y, my - x)
End Select

End Function

Private Sub FrontBtn_Click()

    If Check(0, 1) = 0 Then
        Select Case mh
            Case 0
                my = my - 1
            Case 1
                mx = mx + 1
            Case 2
                my = my + 1
            Case Else
                mx = mx - 1
        End Select

        Call Draw

        If Check(0, 1) = 2 Then
            MsgBox "ゴール" & Score & "点です。", vbInformation
            TmrMain.Enabled = False
            FrontBtn.Enabled = False
            LeftBtn.Enabled = False
            RightBtn.Enabled = False
            Picture1.Cls
            Startbtn.Enabled = True
            Label1.Visible = True
            Label3.Visible = True

            Call Form_Load
        End If
    End If

End Sub

Private Sub GiveUpbtn_Click()

    MsgBox "ギブアップ!!", vbInformation
            TmrMain.Enabled = False
            FrontBtn.Enabled = False
            LeftBtn.Enabled = False
            RightBtn.Enabled = False
            GiveUpbtn.Enabled = False
            Picture1.Cls
            Startbtn.Enabled = True
            Label1.Visible = True
            Label3.Visible = True

            Call Form_Load
```

```
End Sub
Private Sub TmrMain_Timer()

'経過する時間によって残りタイム減少・点数減少

    Time = Time - 1
    TxtTime.Text = Time
    Score = Score - 1
    Text1.Text = Score

'時間がなくなったら

    If Time < 1 Then
        MsgBox "タイムオーバーです。", vbInformation
        FrontBtn.Enabled = False
        RightBtn.Enabled = False
        LeftBtn.Enabled = False
        TmrMain.Enabled = False
        Picture1.Cls
        Startbtn.Enabled = True
        Label1.Visible = True
        Label3.Visible = True

        Call Form_Load
    End If

End Sub
Private Sub LeftBtn_Click()

    mh = (mh + 3) Mod 4
    Call Draw

End Sub

Private Sub Quitbtn_Click()

    Unload Me
    End

End Sub

Private Sub RightBtn_Click()

    mh = (mh + 1) Mod 4
    Call Draw

End Sub


Private Sub Startbtn_Click()

    TmrMain.Enabled = True
    GiveUpbtn.Enabled = True

    Call Init

End Sub
```

《リスト2》「迷路」のMapdata.datリスト

```
1111111111111111111
1101100011000000021
1000111010011110111
1110101010111001010 1
```

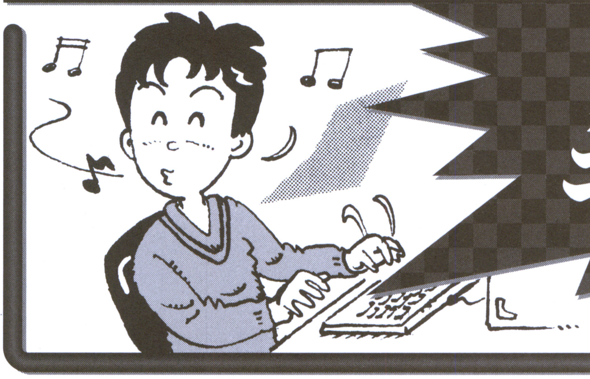

★やった！年末ジャンボ宝くじの1等前後賞が一気に当たったぞ！と言いたいなぁ。さて，結果は…。（愛知県知多郡・**四谷一獲？歳**）……【**影**：夢でもいいから，宝くじ当たってみたいものだ。**つぐ美**：最近，競馬もぜんぜん当たんないしね，影さん。**影**：しーっ。】

Windows(Visual Basic V6.0)用

みんなを驚かせよう

キートリック

山下　慎平（16歳）



内容

メモ帳に見せかけたソフト上でキーを押すと，あらかじめ指定しておいたファイルの内容が，押したキーとは無関係に1文字ずつ現れるソフトです。

使いかた

家や学校，職場などで，あらかじめ使用するテキスト・ファイルの設定をしておきます。そして，友人が来たときを見計らって，さりげなくタイプしはじめます。このとき押すキーは何でも構いませんが，IMEはオフにしておく必要があります。

向こうがディスプレイを覗いて驚いたなら，あなたはこのソフトを正しく使っているといえるでしょう。

注意点

このソフトは，キー入力が速くなったように見せかけるだけです。あくまでも見せかけるだけですので，実際に速くなったと勘違いしないようにしてください。

勘違いしたままワープロ検定特段に挑み不合格だった場合，受験料1万円返せとおっしゃられても，作者の知り得るところではないことをお伝えしておきます。

最後に

実用性ゼロのまったくしょーがないソフトです。しかし作者が学友に試したところ，ほとんどの人がひっかかりました。みなさんもぜひ試してみてください。

効果や感想などをお待ちしています。

E-mail：09065917241@jp-c.ne.jp

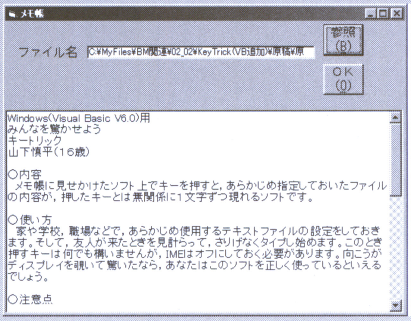
CHECKER FLAG

編：すごく単純な内容ですが，知らない人が見ると意外と引っかかりますよ。こういったジョークの利いた作品もおもしろいですね。

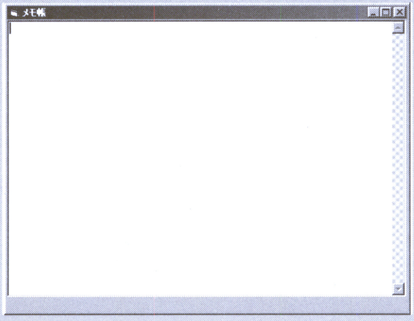
Dr.D：プログラムの本質的な内容は，ファイルの基本操作くらいなので，やはりアイデア勝ちといった作品じゃな。細かなところでは，ファイル指定後にスクロールしながら画面一杯になるテキスト・ボックスの表示にこだわっていると言える。

つぐ美：影さんなんかは，私たちに“仕事しているぞ”と見せかけるために，このソフトを使用しそうね。

影：ギクッ。

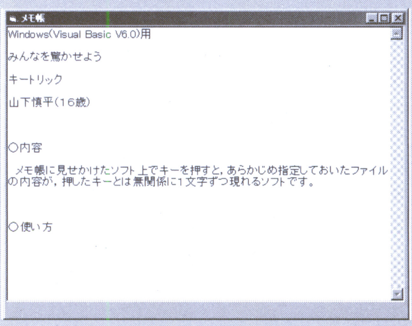


《写真1》まずは，テキスト・ファイルを指定しよう

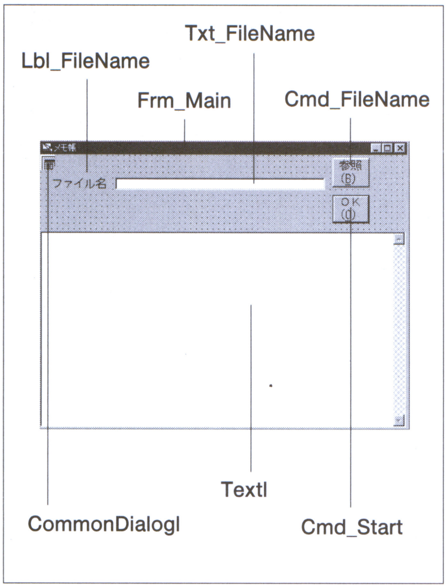


《写真2》このメモ帳もどきの画面でキーを押していくと…

《写真3》キー内容とは関係なく，ファイル内容が表示されていく



★「コナミ流」に続き「ベーセガ」撤退とはBMがピンチです。つかちーもピンチ。（福井県武生市・**つかちー17歳**）……【**影**：誰が○ピン子だ！**編**：ピンチの後にはチャンスが来るってことを知らないな？ベーマガはこれから大きく発展するぞ！応援よろしく！】



《第1図》フォームの配置図

《第１表》変数表

フォームの設定	
Name	Frm_Main
Caption	メモ帳
Font	FixedSys
…Size	13.5
Height	6180
Icon	(ｱｲｺﾝ)
Left	-15
LinkTopic	Form1
ScaleHeight	5775
ScaleWidth	7905
StartUpPosition	2 '画面の中央
Top	-30
Width	8025
コマンドボタンの設定	
Name	Cmd_Start
Caption	ＯＫ(&O)
Default	-1 'True
Height	600
Left	6300
TabIndex	4
Top	840
Width	790
Name	Cmd_FileName
Caption	参照(&B)
Height	615
Left	6300
TabIndex	3
Top	60
Width	790
テキストボックスの設定	
Name	Txt_FileName
Font	ＭＳ Ｐゴシック
…Size	9
Height	250
Left	1620
TabIndex	2
Top	480
Width	4510
Name	Text1
Font	ＭＳ Ｐゴシック
…Size	11.25
Height	4155
Left	0
MultiLine	-1 'True
ScrollBars	2 '垂直
TabIndex	0
Top	1620
Width	7870
ラベルの設定	
Name	Lbl_FileName
Caption	ファイル名
Height	315
Left	240
TabIndex	1
Top	480
Width	1275
コモンダイアログの設定	
Name	CommonDialog1
Left	0
Top	0

「キートリック」のプログラム・リスト

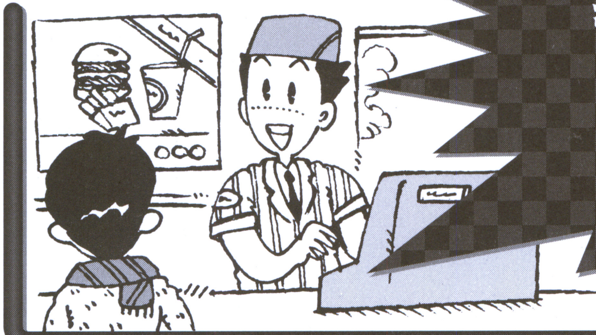
```
Option Explicit
Dim Text, char(10000) As String
Dim FileName, Stroke, Pus, init, TxLen, StrokeNum As Integer

Private Sub Cmd_FileName_Click() 'ファイルを開く
CommonDialog1.FileName = ""
CommonDialog1.Filter = "テキストファイル(*.txt)|*.txt|すべてのファイル(*.*)|*.*"
CommonDialog1.ShowOpen
If CommonDialog1.FileName <> "" Then
    FileName = CommonDialog1.FileName
    Txt_FileName.Text = FileName
    Call Reader  '確認のためにテキストボックスに表示する。
End If
End Sub

Private Sub Cmd_Start_Click()
init = 0  '設定モードではないことを示すフラグ設定
Text1.Text = ""  'テキストボックスの文字を消す。
Txt_FileName.Visible = False
Cmd_FileName.Visible = False
Cmd_Start.Visible = False
Lbl_FileName.Visible = False
If Txt_FileName.Text <> FileName Then  'ファイル名を手入力した場合
    FileName = Txt_FileName.Text
    Call Reader
    Text1.Text = ""
End If
Dim i As Integer
For i = 0 To 1440  'テキストボックスをフォームに合わせる。(視覚効果付)
    Text1.Top = 1440 - i
    Text1.Height = Text1.Height + 1
Next i
TxLen = Len(Text)
If TxLen > 9999 Then  '文字数が多い場合は減らす。
    Text = Left$(Text, 9999)
    TxLen = 9999
End If
For i = 0 To TxLen  '1文字ずつに分ける
    char(i) = Mid$(Text, i + 1, 1)
Next i
End Sub


Private Sub Form_Load()
Stroke = 1: init = 1
FileName = "c:\windows\tips.txt"  'デフォルトのファイル名
Txt_FileName.Text = FileName
Call Reader
End Sub

Private Sub Reader()  'ファイルを読みこむ
Dim works As String
Text = "": works = ""
Open FileName For Input As #1
Do Until EOF(1)
Line Input #1, works
Text = Text + works + vbCrLf
Loop
Close #1
Text1.Text = Text
End Sub

Private Sub Form_Resize()  '大きさを変えたとき
If init = 0 Then
    Text1.Height = Frm_Main.Height - 300
    Text1.Width = Frm_Main.Width - 100
End If
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)  'キーが押されたとき
If init = 1 Then KeyAscii = 0: Exit Sub
If Pus >= TxLen Then Pus = 0
KeyAscii = Asc(char(Pus))
Pus = Pus + 1
End Sub
```

★復帰のきっかけがなくなってしまったなあー。これからは買うための情報源として読むか…それとも買ったらまた帰ってくるか…。なんかちょっと心苦しい。部屋に1人浮いてる感じなんスよね…。(愛知県名古屋市・わいち301　27歳)……【編：うーむ。】

超大作プログラム

リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

Windows（Visual Basic V6.0）用



注文されたメニューを素早く用意

速食（そくしょく）

はりまお（25歳）

物語

時は近未来。あなたは，企業戦士が行き交う街にあるハンバーガー・ショップの店長。ほとんどオートメーション化され，店で働く人間は1人しかいない。

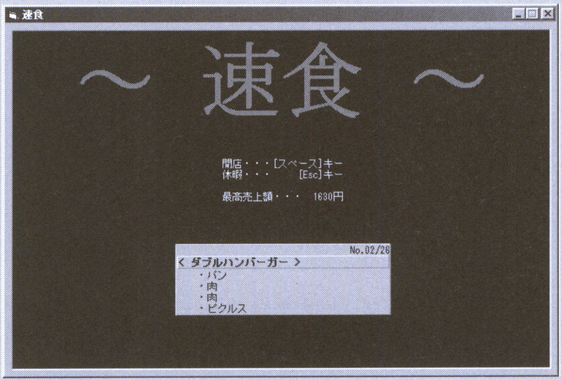
しかし，人間とは無駄が好きな生き物。中途半端なオートメーションが，彼の悩みの種だった。

遊びかた

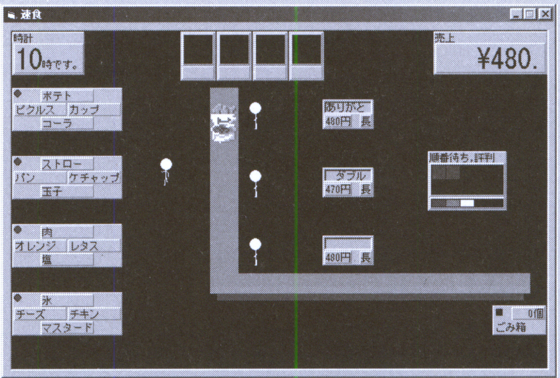
まず，タイトル画面で商品メニューとそれに必要な食材を覚えましょう。結構多いですが，単純な組み合わせなので何とかなると思います（何とかならなかったらゴメンなさい）。営業時間の午前9時から午後9時まで，ミッチリ働いてください。

プレイヤーは，カーソル・キーで動かします。食材は，左側にある食材パネルから選択します。食材の配置は，カーソル・キーに対応しているので目的の食材があるパネルのところへ行き，ほしい食材の方向を押してください。スペース・キーを押せば通路に戻ります。もし，まちがえた場合は，ゴミ箱に捨てることになります（あ〜，もったいない）。

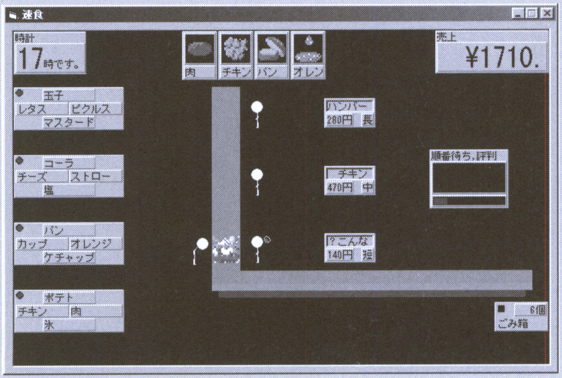
注文品に必要な食材が揃ったらお客



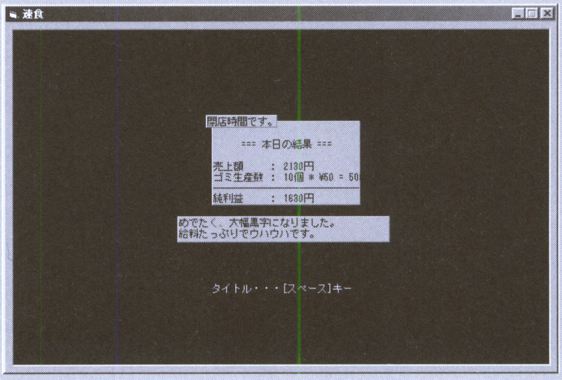
《写真1》タイトル画面でメニューを覚えよう



《写真2》お客様が要望しているメニューを出していく



《写真3》まちがえると罵声をあびるはめに



《写真4》黒字を目指して働くのだ

★1月号にてPIECEにちょっと興味が。投稿プログラムでもPIECEは受け付けるんでしょうか？（青森県南津軽郡・ていこく22歳）……
【編：もちろんです。誰がPIECE投稿一番乗りになるんでしょう。他の機種でもどんどん投稿受け付けます。よろしくね。】

さんのところへ行きます。正しい商品を渡すことができると，お客さんは代金を頂けます（あ〜，ありがたや）。

プレイヤーの働きっぷりが，店の評判に反映され，お客さんの来店数に影響します。“順番待ち，評判”パネルの上側に待ち人数とその心理状態を示しています。イライラが募ると赤くなります。その下に店の評判が表示されます。赤はヤバイです。緑は良い感じです。

改造

画像や食材・メニューデータは，面倒くさいけど変更できます。

食材データは，“食材名，画像番号”の順です。画像番号とは，画像が並んでいる順番です。

メニューデータは，“メニュー名，メニューコード，画像情報”の順番に並んでいます。メニューコードは，メニューを識別するためのコードです。商品を構成する食材の番号（食材データを記述した順番）を16進数1桁で昇順に並べて4文字にします。4文字に足りない分は，“Z”で埋めてください。

画像情報とは，商品を表示するのに必要な画像の番号（画像が並んでいる順番）を下から重ねて表示する順番に16進数2桁で並べただけです。

制作後記

この作品は，昔MSX用に作っていたのですが，処理速度が足りずもう一歩のところでお蔵入りしていました。しかし，一部マシン語で書いていたのでMSX版のほうがある意味で速いという皮肉な結果に…。今思うともう少し何とかなったなぁと反省していたりします。

まぁ，そんなこんなで今回の画像は，MSX版のを2倍に拡大して使ってます。32ドットより16ドットで描いたほうが雰囲気出るし作業効率も桁違いに良いですね。僕だけですか？

欲を言うとお客さんも歩かせたかったなぁ。2001年10月号のヅラChuさんに完全に先を越されてるし（制作期間長すぎやって！！）。

CHECKER FLAG

編：内容を理解するとおもしろいゲームなんですが，やはり難点のほうが目についてしまいます。

つぐ美：その難点を挙げると，「食材の組み合わせを覚えないとゲームがプレイできない」，「メニューが多すぎる」，「動作速度が遅い」といった点かしら。これらは，ゲームとして見ればすぐに目につくものばかり。やはり，ゲーム・バランス調整はしっかり行ってほしいわね。

影：操作性にも，もう少し工夫がほしいなぁ。食材メニューから通路に戻るときに，ついつい感覚で［→］キーを押しちゃうから，操作ミスをしてしまう。

Dr.D：リストが長くなったから超大作というのではなく，リストの長さに見合った内容があってこそ，はじめて超大作と呼べるのではないかな？

Pick Up LIST

《リスト1》変数宣言部分のリスト

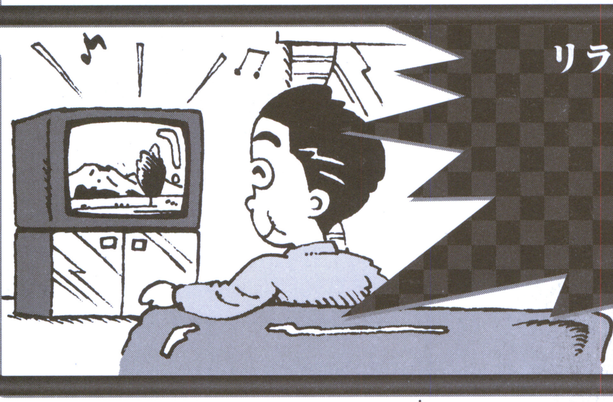
```
Pick Up LIST
Private Type Customer
  Actionf As Byte '行動フラグ(0:注文中 / 1:お礼中 / 2〜5:怒り中 / 255:お客さん無し)
  Order(1) As String '注文台詞と注文品のメニュー登録
  Scrollc As Byte '注文スクロールカウンタ
  Price As Integer '支払額
  Patience As Byte '辛抱強さ
  Actionc As Byte '行動タイミングカウンタ
  Mentalc As Byte '心理変化カウンタ
End Type

'=== 変数宣言 ===
Dim Gf As Byte 'ゲーム状態フラグ（0:タイトル / 1:ゲーム中 / 2:クリア / 3:ゲームオーバー）
Dim Tm(1) As Byte '時間（0=時,1=分）
Dim Fi(InMAX) As Ingredients, Md(MnMAX) As Menu '食材情報, メニュー情報
Dim Io(InMAX) As Byte '食材を表示する順
Dim Pm(2, 3, 1) As Integer, Pp(1) As Byte 'プレーヤーの移動座標, プレイヤーの現在地（移動座標配列の要素数）
Dim CDf As Byte, ps As Long, Hs As Long '接客&ごみ箱モードフラグ, 売上額, 最高売上額
Dim Cn As Byte '押されたカーソル番号(MSX-BASIC:STICK関数準規準)
Dim Iso As String, Ic As Byte, Dc As Integer '選択された食材によるメニュー構成データ, 食材を選んだ数, 食材を捨てた数
Dim Cd(2) As Customer, Wt As Integer, Wm(10) As Byte, Rv As Integer 'お客さん情報, 順番待ち人数, 順番待ち客の心理, お店の評判
```

Dr.Dのワンポイント解説

Dr.D：今回はゲーム内の処理でなく，プログラムでは基本的な変数宣言部分のリストを掲載してみた。というのも，このリストは変数宣言のコメントがしっかり記述されていたからじゃ。最近のベーマガHPのプログラム掲示板などを見ていると，このプログラムの基本である変数をうまく扱いきれていない人が多いように思う。このリストを見ることで，ある処理を行うのにどのような感じで変数を扱っているのかをしっかり認識してもらいたいものじゃ。

★高校入試を考えただけで，今まで勉強してきたことが忘れてしまいます。僕，プレッシャーに弱いんです。こんな僕におまじないよろしくお願いします。影さん。（福井県福井市・受験生2号15歳）……【影：ちちんぷいぷい，痛いの痛いのとんでいけー。どう？】

Windows(VBScript+DirectX)用



リラクゼーション・プログラム？！

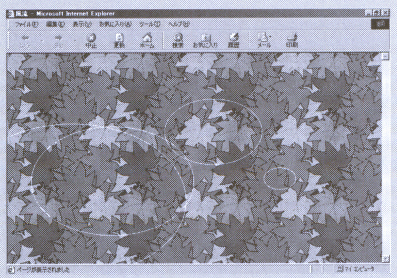
風流（ふうりゅう）

黒田　国男（40歳）

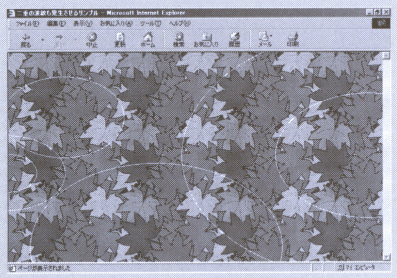
内容

色とりどりの落ち葉が散る水面に，一つまた一つと雨が落ちては波紋を広げて消えていきます。この水の表面の風流なさまを，デジタルな楕円のアニメーションで表現してみました。実際の水の上に起こる波紋と見比べると，写実的なリアルさに欠けているかもしれません。が，この波紋の抽象化されたイメージは，パソコンの画面上に広がる無色透明な水面に，はっきりと浮かび上がってくるように見えます。

ゲームではありませんが，つい見入ってしまうようなプログラムに仕上がりました。



《写真1》疲れたときに見るのがオススメ？



《写真2》いろんな改造にも挑戦してみよう

プログラムの仕組み

プログラムのしくみそのものは，いたって簡単です。DirectAnimationの初期化からコントロールにオブジェクトをセットしてスタートするまでの間に，つぎのような処理を行っています。

①動的に変化する乱数値オブジェクトを生成して，円の発生場所を不特定にする。
②任意の位置から現れて広がり消える楕円のイメージを生成する。
③背景イメージを設定する。
④ ③の背景イメージ上に，②で生成した楕円のイメージを周期的に重ねて，複数の波紋のイメージを生成する。

この中でとくに苦心した個所は，①の部分です。私自身の勉強不足のせいで，ランダムな変化をDirectAnimationに組み込む方法がわかりませんでした。やむなく30余りのテスト・プログラムをVBScriptで作って実験しました。そんな試行錯誤を繰り返した末に，乱数値オブジェクトを生成する関数が出来上がりました。

このDefRandomRange関数は，一般的な乱数値発生関数とするには，まだ70％の完成度しかありませんが，今回のプログラムをイメージどおりに動かすのにはこれで十分でした。

なお，プログラムの基本部分には，マイコンBASIC Magazine 2001年9月号，10月号の『IE＋DirectAnimationマルチメディア・プログラミング』記載のサンプルを流用させていただきました。

また，使用している画像は，Windowsに標準で付属しているBMP画像ファイル“Leaves.bmp”です。

あとがき

Windowsデスクトップ上で手軽に作って動作確認ができるのがVBScriptとDirectAnimationの良いところなのでしょう。この利点を生かしてプログラムをあれこれいじったり組み合わせていたら，いろんな改造サンプルを作ってしまいました。

★参考にしたもの
・マイコンBASIC Magazine 2001年9月号と10月号の『IE＋DirectAnimationマルチメディア・プログラミング』
・Microsoftホームページ(英語版)のMSDN LibraryのDirectAnimationの関数およびプロパティのReference

CHECKER FLAG

影：基本的なDirect Animationプログラムだけど，たまにはこんな作品もいいかもしれないなぁ〜。

編：内容は単純ですけど，説明の文章を読んでから実行すると水面をイメージしてしまいますね。作ったプログラムに対する説明も大切，という良い見本ですね。

Dr.D：別表を参考にして改造にも挑戦してほしいものじゃな。また，Direct Animationプログラムでは数値 の扱いかたにチョットコツがいるので，慣れていないと，たとえば今回のような乱数発生という単純な処理でも意外と苦戦してしまう。やはり過去の投稿作品やサンプルの処理をしっかり研究するのがオススメじゃな。

《第1表》改造点または改造の例

その1「真上から見た波紋にする」

これは余りに簡単です。楕円を設定しているOval関数の二つの引数を同じ値にするだけです。

```
Set cycle = m.Oval(30,20).Draw(border)   '←楕円のイメージ
Set cycle = m.Oval(30,30).Draw(border)   '←円のイメージ
```

その2「背景を無地の色にする」

背景イメージを設定している二行，すなわち

```
Set back = m.ImportImage("Leaves.bmp")
Set back = back.Tile
```

を次の一行に変更します。

```
Set back = m.SolidColorImage(m.Black)
```

その3「波紋の広がる大きさとスピードを変える」

以下に掲げる定数の値を変更することによって，波紋の広がる大きさとスピードを変えることができます。

・波紋自動発生の間隔（B_INTERVAL）
何秒間隔で波紋を発生させるかを指定します。
(SCALE_TIME/B_INTERVAL)個の波紋が同時に表示されているように見えます。
B_INTERVALの値を小さくすると，画面上に表示される波紋の数が増えます。

・最大拡大倍数（MAX_SCALE）
波紋の広がる大きさを指定します。Oval関数で最初に設定した円または楕円の何倍の大きさまで大きくするかをその倍数で規定します。
MAX_SCALEの値を小さくすると，波紋の広がる範囲が小さくなります。
その時，波紋の広がるスピードは遅くなります。

・波紋の拡大に要する時間（SCALE_TIME）
一つの波紋が発生して消えるまでの時間を秒単位で指定します。
SCALE_TIMEの値を小さくすると，一つの波紋の表示される時間が短くなります。
その代わり，波紋の広がるスピードは速くなります。

これらの定数の変更を組み合わせることにより，変わったパターンを作ることが出来ます。一例を挙げますと，B_INTERVALを0.4に，MAX_SCALEを90に，SCALE_TIME を2にしますと，残像現象を利用して複数の波紋を発生させているように見せることができます。

その4「マウスクリックした場所にも波紋を発生させる」

少々手間がかかりますが，以下の追加・変更で実現できます。

①Window_onLoad関数に次の変数の宣言を追加します。

```
Dim mouse    'As DAPoint2    'マウスポインタの座標(x, y)
Dim cycle1   'As DAImage     '円のイメージ（自動発生用）
Dim cycle2   'As DAImage     '円のイメージ（マウスクリック用）
Dim ripple1  'As DAImage     '波紋のイメージ（自動発生用）
Dim ripple2  'As DAImage     '波紋のイメージ（マウスクリック用）
```

②DirectAnimationの初期化，すなわち

```
Set m = DAControl.PixelLibrary
```

の後で，次の処理を追加します。

```
Set mouse = m.MousePosition
Set mouse = m.UntilEx(mouse, m.Always.Snapshot(mouse))
```

③広がる円のイメージを生成する個所，すなわち

```
Set cycle = cycle.Transform(m.Translate2Anim(RndX, RndY))
```

を次のように変更します。

```
Set cycle1 = cycle.Transform(m.Translate2Anim(RndX, RndY))
Set cycle2 = cycle.Transform(m.Translate2Point(mouse))
```

④波紋のイメージを生成する個所全体を③に合わせて次のように変更します。

```
Set ripple1 = CreateObject("DirectAnimation.DAImage")
ripple1.init m.Until(back, _
                     m.Timer(B_INTERVAL), _
                     m.Overlay(cycle1, ripple1))
Set ripple2 = CreateObject("DirectAnimation.DAImage")
ripple2.init m.Until(m.EmptyImage, _
                     m.LeftButtonDown, _
                     m.Overlay(cycle2, ripple2))
Set ripple = m.Overlay(ripple2, ripple1)
```

以上の追加・変更により，左クリックした個所にも波紋が発生可能になります。
遊び方としては，マウスをあちこち動かしてクリックしたり，一ヶ所で連続してクリックしたりします。

その5「二重の波紋も発生させる」

Window_onLoad関数に次の変数の宣言を追加します。

```
Dim cycle0    'As DAImage    '円の基本イメージ
```

また，楕円を設定している行，すなわち

```
Set cycle = m.Oval(30,20).Draw(border)
```

を次のように変更します。

```
Set cycle0 = m.Oval(30,20).Draw(border)
Set cycle = CreateObject("DirectAnimation.DAImage")
cycle.init m.Cond(m.GTE(DefRandomRange(2), m.DANumber(0)), _
                  cycle0, _
                  m.Overlay(cycle0.Transform(m.Scale2Uniform(2)), cycle0))
```

その6「カラフルな色で波紋を発生させる」

改造点その１・その２と共に，Window_onLoad関数に次の変数の宣言を追加します。

```
Dim RndC 'As DAnumber     'ランダムな色指定
Dim hsl  'As DAColor      'DAColorオブジェクト
```

動的に変化する乱数値として次の一行を追加します。

```
Set RndC = DefRandomRange(256)
```

また，円の境界線を設定している個所，すなわち

```
Set border = m.DefaultLineStyle.Color(m.White).Width(1)
```

を次のように変更します。

```
Set hsl = m.ColorHslAnim(RndC, m.DANumber(1), m.DANumber(0.5))
Set border = m.DefaultLineStyle.Color(hsl).Width(1)
```

その7「虹色に変化する波紋を発生させる」

改造点その１・その２と共に，Window_onLoad関数に次の変数の宣言を追加します。

```
Dim RndC 'As DAnumber     'ランダムな色指定
Dim hsl  'As DAColor      'DAColorオブジェクト
```

動的に変化する乱数値として次の一行を追加します。

```
Set RndC = DefRandomRange(256)
```

また，円の境界線を設定している個所，すなわち

```
Set border = m.DefaultLineStyle.Color(m.White).Width(1)
```

を次のように変更します。

```
Set num = m.Add(RndC, m.Mul(m.LocalTime, m.DANumber(0.1)))
Set hsl = m.ColorHslAnim(num, m.DANumber(1), m.DANumber(0.5))
Set border = m.DefaultLineStyle.Color(hsl).Width(1)
```

色の変化をゆっくり観察できるようにするために波紋の動きをコントロールする定数を次のように変更します。

```
Const B_INTERVAL = 1.6     '波紋自動発生の間隔
Const MAX_SCALE = 20       '最大拡大倍数
Const SCALE_TIME = 8       '拡大に要する時間
```

※なお，以上の改造点を加えたサンプルのVBScriptをおまけでつけました。
ダウンロードファイルのSamplesフォルダをご覧ください。

「風流」のプログラム・リスト

```
<HTML>
<TITLE>風流</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//-----------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Kuroda Kunio
'//
'//    『風流』
'//
'//-----------------------------------------------------------------
Option Explicit

  Dim m          'As DAStatics     'DAStatics

Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Const B_INTERVAL = 0.8              '波紋自動発生の間隔
  Const MAX_SCALE = 18                '最大拡大倍数
  Const SCALE_TIME = 3                '拡大に要する時間
  Dim RndX 'As DANumber           'ランダムなX座標
  Dim RndY 'As DANumber           'ランダムなY座標
  Dim border    'As DALineStyle    '線のスタイル
  Dim cycle     'As DAImage        '円のイメージ
  Dim num  'As DANumber           'DANumber
  Dim ripple    'As DAImage        '波紋のイメージ
  Dim back      'As DAImage        '背景イメージ
  '//
  '// DirectAnimationの初期化 (ピクセル座標系)
  '//
  Set m = DAControl.PixelLibrary
  '//
  '//      円の発生場所を不特定にする
  '//
  Randomize
  Set RndX = DefRandomRange(Document.Body.ClientWidth)
  Set RndY = DefRandomRange(Document.Body.ClientHeight)
  '//
  '// ひろがる楕円のイメージ (任意の位置から現れて自然に消える)
  '//
  Set border = m.DefaultLineStyle.Color(m.White).Width(1)
  Set cycle = m.Oval(30,20).Draw(border)
  Set num = m.SequenceArray(Array(m.Interpolate(1, MAX_SCALE, SCALE_TIME), _
                                  m.DANumber(0).Duration(1)))
  Set cycle = cycle.Transform(m.Scale2UniformAnim(num))
  Set cycle = cycle.Transform(m.Translate2Anim(RndX, RndY))
  '//
  '// 背景イメージ
  '//
  Set back = m.ImportImage("Leaves.bmp")
  Set back = back.Tile                          '← タイリング処理
  '//
  '// 波紋のイメージ
  '//
  Set ripple = CreateObject("DirectAnimation.DAImage")
  ripple.init m.Until(back, _
                      m.Timer(B_INTERVAL), _
                      m.Overlay(cycle, ripple))
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = ripple
  DAControl.Start
  Call Window_onResize()
End Sub

'-----------------------------------------------------------------
' 動的に変化する乱数値オブジェクトを生成する
'-----------------------------------------------------------------
Function DefRandomRange(Range)
  Dim num  'As DANumber       'DANumber

  Set num = m.Mul(m.SeededRandom(Rnd), m.DANumber(Range))
  Set num = m.Sub(num, m.DANumber(Range ¥ 2))   '←座標系対処の処理
  Set DefRandomRange = m.InterpolateAnim(num, num, m.DANumber(-1))
End Function

'-----------------------------------------------------------------
' ウィンドウのサイズ変更時の処理
'-----------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

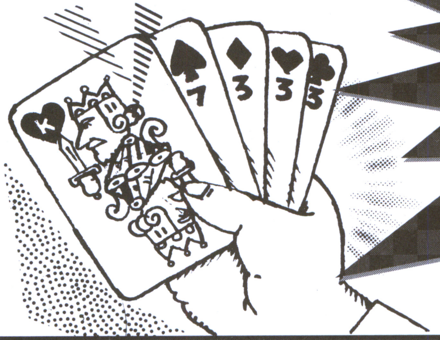
★今月は学校のパソコンから投稿しようと思いましたが，先生に見つかったので，家のパソコンからの投稿となりました。来月こそは隙を見て…。（愛媛県東宇和郡・**毛利祐允18歳**）……**【影：友達みんなで，ぜひ挑戦してもらいたい。じつは，先生も隠れて…。】**

超大作プログラム　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(VBScript+DirectX)用

ここまで実現！DirectAnimationゲーム

スピード



金井　義男（39歳）

内容・遊びかた

CPUと対戦するトランプ・ゲームのスピードです。マウスでカードをドラッグ&ドロップして置いていきます。

操作方法

下側左の裏返しカードがスタートボタンになります。まずは，マウスでクリックしてください。

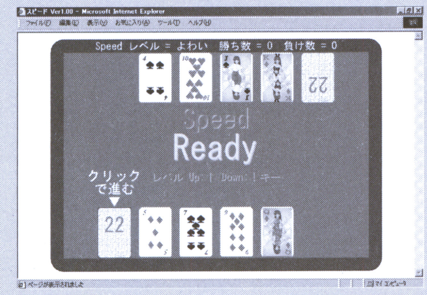
トランプの置きかたはカードをマウスでドラッグ&ドロップします。空いたところにつぎのカードを置くには，スタートボタンをクリックしてください。

カードが置けなくなってゲームの進行がストップしたら，やはりスタートボタンをクリックしてください。

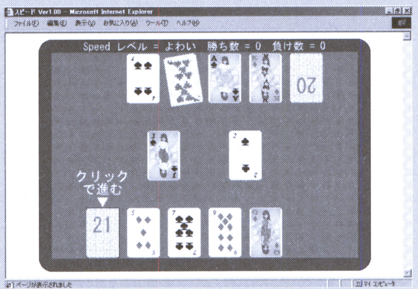
自分の裏返しカードが0になっても，相手のカードが0になるまではスタートボタン(手持ちカード部が黄色表示されます)を押してゲームを進行させます。プレイヤーとCPU両方の裏返しカードがなくなったら，自分のカードから1枚を選んで右側へ置いてください。勝てばCPUのレベルがアップしますし，負ければCPUのレベルはダウンします。

プログラムの入力方法と注意点

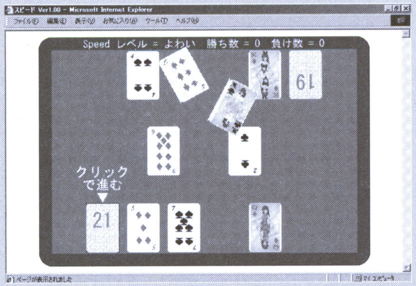
メモ帳を使って入力，改造ができますが，デバッガとテキスト・エディタ機能のあるツールを使用することをおすすめします。



《写真1》ブラウザで「スピード」を体験



《写真2》カード表示の演出面も要チェック



《写真3》マウスで素早くカードを置いていく

《写真4》勝敗結果もしっかり表示される

CHECKER FLAG

編：これはスゴイです。DirectAnimationでもここまで本格的なゲームが制作できるんですね。

つぐ美：COMのレベルは，カードを置く速さのちがいで表しているのね。一番強いモードは，チョット速すぎるかなっと思いますが，強さがしっかりとわけられているのは，なかなかのものじゃないかしら？

Dr.D：「スピード」というゲームは，比較的簡単な内容じゃが，VBScript＋DirectXでこういったゲームのプログラムをするのは，かなり大変なのじゃ。終了部分でチョット変わった手順を踏まなければいけないのも，こうした理由からじゃろう。逆にカードを回転させるなどといった演出面の処理は，手軽にできたりするのじゃがな。

★DVD-ROMドライブ買いました。でも，CD-RWとのコンボドライブとかいうのが今は主流なんですね。買ってから気がつきました。安かったはずだ。お店の人，なんで教えてくれなかったの？（宮城県仙台市・**タイピングZZ19歳**）……**【影：お金なさそうだったから？】**

●改造のアドバイスと苦労した点●

CPU側がカードを置いたとき，下側のカードが一瞬再表示されてしまうのが，どうしてかわかりませんでした。ここが難点です。プログラムが長くなり，全体がわかりにくく，バグやどこが悪いのか見つけるのに苦労しましたが，ひたすらコメントをいれたり，サブ・ルーチンにわけたりしてなんとかできました。

ベーマガではボツになったものや，バージョン・アップしたものを私のホームページでも紹介していますので暇のある方は見にきてください。

http://www.d7.dion.ne.jp/~kanaisun

Pick Up LIST

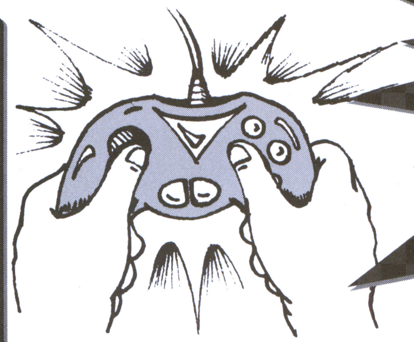
《リスト1》COM処理部分のリスト

```
●Pick Up LIST

《リスト1》COM処理部分のリスト

'// ----------------------------------------------
'// カードに対応するコンピュータ対応処理
'// ----------------------------------------------
Private Sub Computer()
  Dim numX          'As DANumber      カードを元に戻すための変動値
  Dim numY          'As DANumber      カードを元に戻すための変動値
  Dim okflag     'As Boolean     CPU側が置けるカード中に同じNOのカードがあるか判定
  Dim i, j, k      'As Long ループカウンター

  '// セッターカードとのチェックして置けるカードの中に同じNOのカードがあるか判定
  For i = 0 To 1
    For j = 0 To 2
      If ctrn(i) Mod 13 = (cpun(j) + 1) Mod 13 And cpuf(j) = True Or _
         ctrn(i) Mod 13 = (cpun(j) + 12) Mod 13 And cpuf(j) = True Then
        For k = j + 1 To 3
          If cpun(j) Mod 13 = cpun(k) Mod 13 And _
             cpuf(j) = True And cpuf(k) = True Then
            okflag = True
            Exit For
          End if
        Next
      End If
      If okflag = True Then Exit For
    Next
    If okflag = True Then Exit For
  Next
    If okflag = False Then
      '// 手持ちカードとセンターカードを比較して置けるものあるか判定
      For i = 0 To 1
        For j = 0 To 3      '// セッターカードとのチェック
          If ctrn(i) Mod 13 = (cpun(j) + 1) Mod 13 And cpuf(j) = True Or _
             ctrn(i) Mod 13 = (cpun(j) + 12) Mod 13 And cpuf(j) = True Then
            okflag = True
            Exit For
          End If
        Next
          If okflag = True Then Exit For
      Next
    End If
    If okflag = True Then
      '// カードを回転させがら置く
      Set numX = m.Interpolate(80 * j - 120, 160 * i - 80, pas_sec)
      Set numY = m.Interpolate(-150, 0, 2)
      cmx(j).SwitchTo numX
      cmy(j).SwitchTo numY
      crot(j).SwitchTo m.Sequence(m.Interpolate(0, 2 * PI, _
                pas_sec).Duration(pas_sec), m.DANumber(0))
      chkcnt(j) = 0          '// カードを置きに行っている間のカウンター
      chkcno(j) = i          '// センターカード左右NO(0 or 1)
    End If
End Sub
```

Dr.Dのワンポイント解説

Dr.D：本来if文などによる条件分岐が難しいDirectAnimationじゃが，この作品はそんなことを感じさせないくらい，しっかりゲームとして成り立っている。この部分も，if文やfor文内での細かな記述が難しいVBScript+DirectX環境において，判断用のフラグを用意するなど様々な工夫を凝らしてループ処理を実現しておるのがわかる。その分，Exit命令が多いなどで多少汚く見えるかもしれんが，まあこれは仕方ないじゃろうな。

Windows（Visual C++ V6.0）用



手軽にゲーム・コントローラを利用しよう

BCJOY（ビーシージョイ）クラス&サンプル

BlueCube（19歳）

内容

ジョイパッドを使いたくなったので作ってみました。やっぱクラス化すると便利だからね。

みなさん，これ使ってプログラムの完成度を少し上げてください。

使いかた

まずはプログラムのフォルダにbcjoyc.h（リスト1）を入れます。ソース内では，これをインクルードしてください(#include"bcjoyc.h")。

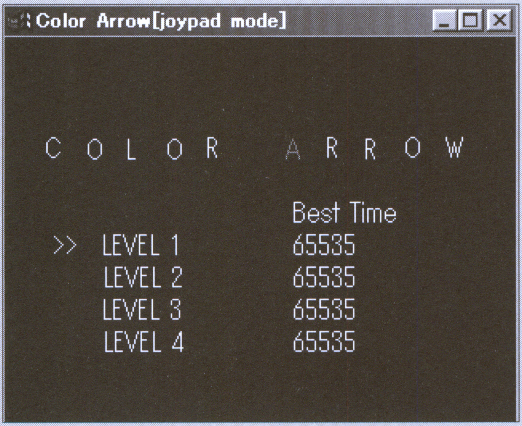
つぎにグローバル変数として，このクラスを宣言します（BCJOY *joy;)。そしてWM_CREATEで,joy=new BCJOY(1)；と入れてください。() の中にはJoyPadのナンバー（1か2）を入れてください。それから，WM_DESTROYで必ず開放（delete joy;）してやってください。

そのあとの使いかたはbcjoyc.hファイル内に記載しておきます。

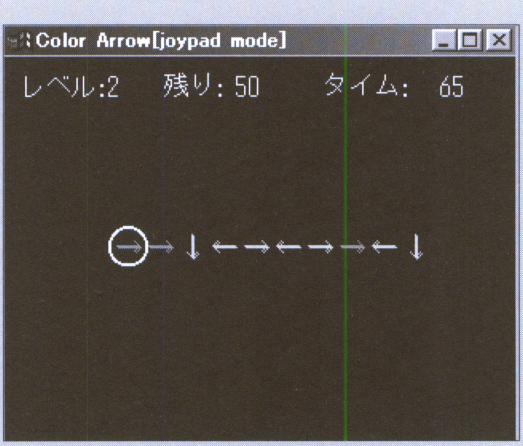
特徴

まず，1番の特徴は簡単に使えること。でも，その分柔軟性はあまりないかも。本格的に作ろうとしたらバッファとか付けられるけど，ソースが長くなるから今回はパス。

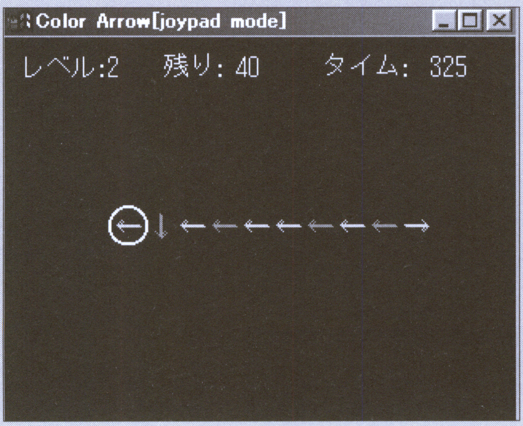
つぎにソースにほとんど手を加えることなくキーボードにも対応できること。他にもいろいろあるけど，とくにはこんなもん。



《写真1》bcjoyクラスを使ったサンプル・ゲーム



《写真2》方向キー＋ボタンでプレイする



《写真3》色に対応したボタン＋方向キーを押せ

《写真4》けっこうハマルゲーム。こんな作品が手軽に作れるゾ

★愛子…いい名だ。私のパソコンにも愛子がっ！ああ愛子ンッ！信じてほしいのだっ！僕の愛子ンッ！！（愛知県名古屋市・村上哲也29歳）……【影：このネタもらった。Dr.D：？？ようわからん。影：Dr.のパソコンにも愛子ンがいるでしょ。Dr.D：あ・い・こ・ん？】

サンプル・プログラム

前回作ったBCBCクラスではへボへボサンプルだったので，今回，少しは完成度のあるゲームにしました。でもソースは短いですが，かなり汚い作りになっています。すみません（なんせテストの真っ只中で作ってるんで…）。

ちなみにJoyPadは4ボタン以上のを使用してください。JoyPadがないときは，「F1」キーでジョイパッドとキーボードの操作モードを変更してください。

遊びかたは，起動するとタイトルが出るので，上下でレベルを選んでボタンを押してください。ゲームがはじまると，色の付いた矢印が一列に並んで出てきます。丸の中の矢印に注目して矢印の方向を押しながら，各色に対応するボタンを押してください（①：緑，②：赤，③：青，④：黄）。

ボタンと方向が矢印と合っていれば矢印が消え，一つ左にずれます。矢印を50個消せばクリアで，そのタイムを競うゲームです。レベルは出現する色の数に対応しています。

プログラムの仕組み

bcwin.h（**リスト2**）には，あると結構便利な関数が入ってます。bcjoyc.hはBCJOYクラスが入ってます。main.cpp（**リスト3**）はメインのプログラムです。

改造のアドバイスと苦労した点

今回，じつは作りはじめて10分で挫折しました（笑）。winmm.libを読み込んでなかったのが原因だったんだけど，夕方にハッと気が付いてクラス自体はその日に完成させました。

で，悩んだのがどんなサンプル・ゲームにするか？です。外部ファイルを使わずにソースを短く，ほどほど遊べるゲームを考えるのはむずかしいっす。ソースを載せてほしいため，超大作にはしたくないので。

このクラスを使った投稿が1冊に二つくらい載って，まとめてソースを載せてもらうのがひそかな野望だったりします(爆)。

つぎはBCPLAYクラスを作るんでお楽しみに♪

♪♪
♪♪♪♪
♪♪

CHECKER FLAG

Dr.D：ジョイパッドといったゲーム・コントローラの使用には，DirectInputを利用している人が多いと思うが，やはり利用するための初期化の記述が面倒なのも事実。このクラスを使えば，手軽にゲーム・コントローラが利用できるので，DirectXなどに慣れていないプログラム初心者や，チョットしたデバッグ用のプログラム操作などに向いていると言えるじゃろう。

影：リストが長くなるという理由で付けていないようだけど，トリガ設定機能もあると，ゲーム・プログラムでの利用の幅が広がったと思うなぁ。

《リスト1》「BCJOYクラス」bcjoyc.hのリスト

```
/*		BCJOYクラスの使い方
	BCJOY(int num)
		int num :ジョイスティックID。1か2のどちらかを入れてください。
		コンストラクタ
	int dir()
		7 8 9			方向を返します。
		4 ・ 6			値はテンキーの5を中心に考えてその周りの8方向の値を返します。
		1 2 3			例）左上=7	下=2	右=6	右上=9
	int dx(int w)
		int w :返り値の大きさを指定。
	ジョイパッドを左に入れた時「-w」右に入れた時「+w」ニュートラル状態で「0」をかえします。
		「w」を省略すると「-1」「0」「+1」を返します。
	int dy(int w)
		int w :返り値の大きさを指定。
	ジョイパッドを上に入れた時「-w」下に入れた時「+w」ニュートラル状態で「0」をかえします。
		「w」を省略すると「-1」「0」「+1」を返します。
	bool button(byte n)
		byte n :ボタン番号を指定。
		指定されたボタン番号のボタンが押されていると「true」を返します。
	int buttonC()
		ジョイパッドの押されているボタンの数を返します。
	void SetJoyMode()
		ジョイスティックモードに切り替えます。
	void SetKeyMode()
		キーボードモードに切り替えます。
	void ChangeMode()
		ジョイスティックモードとキーボードモードを切り替えます。
	void KeySetDir(int d[], int size)
		int d[]	: この配列に（下左右上）の順に指定する仮想キーコードを指定してください。
		int size	: 必ず「4」を指定してください。
		キーボードモード時の方向の対応キーを設定します。
	void KeySetButton(int b[], int size)
		int d[]	: この配列にボタン㈰～の順に指定する仮想キーコードを指定してください。
		int size	: 必ず「8」を指定してください。
		キーボードモード時のボタンの対応キーを設定します。
	その他の注意点
		初期設定時はキーボードの設定は「上下左右」→方向キーの上下左右
			ボタン㈰～㈧が順に「ASDCQWER」と設定されています。
*/

#pragma comment(lib,"winmm.lib")

#define J_UP		(ji.wYpos<=ymin)
#define J_DOWN		(ji.wYpos>=ymax)
#define J_LEFT		(ji.wXpos<=xmin)
#define J_RIGHT		(ji.wXpos>=xmax)

#define K_UP		( (GetKeyState(kd[3])>>8)&0x1 )
#define K_DOWN		( (GetKeyState(kd[0])>>8)&0x1 )
#define K_LEFT		( (GetKeyState(kd[1])>>8)&0x1 )
#define K_RIGHT		( (GetKeyState(kd[2])>>8)&0x1 )

class BCJOY
	{
	private:
		JOYCAPS jc;
		JOYINFO ji;
		int xmin,xmax,ymin,ymax;
		int jid;
		bool mode;//T:joypad F:keybord
		int kd[4],kb[8];
	public:
		BCJOY(int num);
		int dir();
		int dx(int w=1);
		int dy(int w=1);
		bool button(byte n);
		int buttonC();
		void SetJoyMode() {mode=true;};
		void SetKeyMode() {mode=false;};
		void ChangeMode() {mode=~mode;};
		void KeySetDir(int d[], int size);
```

★定期購読を続けたいのですが，最近，申し込み用紙が付いていません。もしかして，ベーマガはなくなってしまうの？（山口県宇部市・守永宜弘20歳）……【**編：**影さんはどうかわかりませんが，ベーマガは，これからますます伸びていきます。応援よろしく！】

```
			void KeySetButton(int b[],int size);
		};

BCJOY::BCJOY(int num)
		{
		jid=(num==1)?JOYSTICKID1:(num==2)?JOYSTICKID2:0;
		joyGetDevCaps(jid,&jc,sizeof(jc));
		xmin=(int)((jc.wXmax-jc.wXmin)*0.25);
		xmax=(int)((jc.wXmax-jc.wXmin)*0.75);
		ymin=(int)((jc.wYmax-jc.wYmin)*0.25);
		ymax=(int)((jc.wYmax-jc.wYmin)*0.75);
		mode=true;
		kd[0]=VK_DOWN;
		kd[1]=VK_LEFT;
		kd[2]=VK_RIGHT;
		kd[3]=VK_UP;
		kb[0]=0x41;kb[1]=0x53;kb[2]=0x44;kb[3]=0x43;
		kb[4]=0x51;kb[5]=0x57;kb[6]=0x45;kb[7]=0x52;
		}

int BCJOY::dir()
		{
		if(mode)
			{
			joyGetPos(jid,&ji);
			return (J_UP)?( (J_LEFT)?7:(J_RIGHT)?9:8 ):(J_DOWN)?( (J_LEFT)?1:(J_RIGHT)?3:2 ):(
(J_LEFT)?4:(J_RIGHT)?6:5 );
			}
		else
			return (K_UP)?( (K_LEFT)?7:(K_RIGHT)?9:8 ):(K_DOWN)?( (K_LEFT)?1:(K_RIGHT)?3:2 ):(
(K_LEFT)?4:(K_RIGHT)?6:5 );
		}


int BCJOY::dx(int w)
		{
		if(mode)
			{
			joyGetPos(jid,&ji);
			return (J_LEFT)?-w:(J_RIGHT)?w:0;
			}
		else
			return (K_LEFT)?-w:(K_RIGHT)?w:0;
		}

int BCJOY::dy(int w)
		{
		if(mode)
			{
			joyGetPos(jid,&ji);
			return (J_UP)?-w:(J_DOWN)?w:0;
			}
		else
			return (K_UP)?-w:(K_DOWN)?w:0;
		}

bool BCJOY::button(byte n)
		{
		if(n<1 && n>jc.wMaxButtons)
			return false;
		if(mode)
			{
			joyGetPos(jid,&ji);
			return ( (ji.wButtons>>(n-1)) & 0x1 );
			}
		else
			return ( (GetKeyState(kb[n-1])>>15)&0x1);
		}


int BCJOY::buttonC()
		{
		int a=0;
		if(mode)
			{
			joyGetPos(jid,&ji);
			for(int i=0;i<jc.wMaxButtons;i++)
				a+=( (ji.wButtons>>i)& 0x1 );
			}
		else
			for(int i=0;i<8;i++)
				a+=( (GetKeyState(kb[i])>>15)& 0x1 );
		return a;
		}

void BCJOY::KeySetDir(int d[],int size)
		{
		if(size>4) return;
		for(int i=0;i<size;i++)
			kd[i]=d[i];
		return;
		}

void BCJOY::KeySetButton(int b[],int size)
		{
		if(size>8) return;
		for(int i=0;i<size;i++)
			kb[i]=b[i];
		return;
		}
```

《リスト2》「BCJOYクラス」bcwin.hのリスト

```
void SetClientSize(HWND hWnd,int x,int y);
void SetCenter(HWND hWnd);
void CreateCDC(HWND hWnd,HDC &hMemDC,int w,int h);

void SetClientSize(HWND hWnd,int x,int y)
		{
		RECT crc,wrc;
		int ww,wh,cw,ch,w,h;
		GetWindowRect(hWnd,&wrc);
		GetClientRect(hWnd,&crc);
		ww=wrc.right-wrc.left;
		wh=wrc.bottom-wrc.top;
		cw=crc.right-crc.left;
		ch=crc.bottom-crc.top;
		w=ww+x-cw;
		h=wh+y-ch;
		MoveWindow(hWnd,wrc.left,wrc.top,w,h,true);
		return;
		}

void SetCenter(HWND hWnd)
		{
		HWND hDWnd;
		RECT drc,rc;
		int x,y,w,h;
		hDWnd = GetDesktopWindow();
		GetWindowRect(hDWnd,&drc);
		GetWindowRect(hWnd,&rc);
		x=(drc.right-(rc.right-rc.left))/2;
		y=(drc.bottom-(rc.bottom-rc.top))/2;
		w=rc.right-rc.left;
		h=rc.bottom-rc.top;
		SetWindowPos(hWnd,HWND_TOP,x,y,w,h,SWP_SHOWWINDOW);
		return;
	}

void CreateCDC(HWND hWnd,HDC &hMemDC,int w,int h)
		{
		void *t;
		BITMAPINFO *bmi;
		char *pBit;
		HBITMAP hbmp;
		bmi=(BITMAPINFO*)malloc(sizeof(BITMAPINFO));
```


★お年玉が入ったら，MSX2＋を買うぞー！！（千葉県山武郡・**野潤上15歳**）……**【Dr.D：**なぜ，MSX2＋なんじゃ？**影：**おっ，いきなり登場のDr.です。**Dr.D：**今月これっきりかもしれないからな。**編集長：**そういう意味じゃ，私もここで登場させていただきます。**】**

```
	ZeroMemory(bmi, sizeof(BITMAPINFOHEADER));
	bmi->bmiHeader.biSize=sizeof(BITMAPINFOHEADER);
	bmi->bmiHeader.biWidth=w;
	bmi->bmiHeader.biHeight=h;
	bmi->bmiHeader.biPlanes=1;
	bmi->bmiHeader.biBitCount=24;
	bmi->bmiHeader.biCompression=BI_RGB;
	t=(void*)pBit;
	hbmp=CreateDIBSection(0,bmi,DIB_RGB_COLORS,&t,NULL,0);
	hMemDC=CreateCompatibleDC(GetDC(hWnd));
	SelectObject(hMemDC, hbmp);
	return;
	}
```

《リスト3》「BCJOYクラス」main.cppのリスト

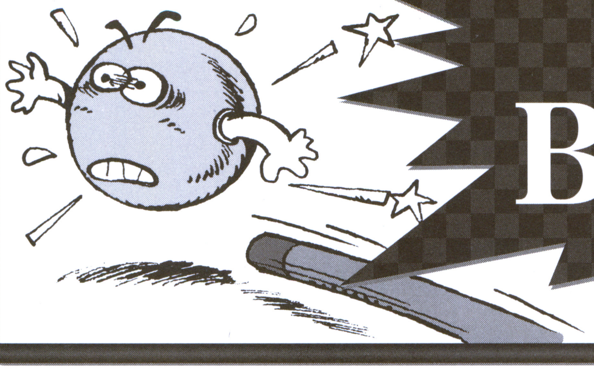
```
#include <windows.h>
#include <time.h>
#include <math.h>
#include "bcjoyc.h"
#include "bcwin.h"

TCHAR szTitle[]="Color Arrow[joypad mode]";
TCHAR szWindowClass[]="WndApp";

ATOM					MyRegisterClass( HINSTANCE hInstance );
BOOL					InitInstance( HINSTANCE, int );
LRESULT CALLBACK	WndProc( HWND, UINT, WPARAM, LPARAM );

HDC				dc;
int				scine;
int				bd[60],bc[60],level;
int				c,t,ht[5];
bool		mode;
BCJOY		*joy;
HPEN		Pen,oldp;
HBRUSH		Brs,oldb;
COLORREF	col;

void	Scine0(HWND hWnd);//タイトル
void	Scine1(HWND hWnd);//ステージ作成
void	Scine2(HWND hWnd);//ゲームループ
void	Scine3(HWND hWnd);//クリア

int APIENTRY WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow)
	{
	MSG msg;

	MyRegisterClass( hInstance );

	if( !InitInstance( hInstance, nCmdShow ) )
		return FALSE;

	while( GetMessage(&msg, NULL, 0, 0) )
		{
		TranslateMessage( &msg );
		DispatchMessage( &msg );
		}

	return msg.wParam;
	}

ATOM MyRegisterClass( HINSTANCE hInstance )
	{
	WNDCLASSEX wcex;

	wcex.cbSize = sizeof(WNDCLASSEX);
	wcex.style			= CS_HREDRAW | CS_VREDRAW;
	wcex.lpfnWndProc	= (WNDPROC)WndProc;
	wcex.cbClsExtra		= 0;
	wcex.cbWndExtra		= 0;
	wcex.hInstance		= hInstance;
	wcex.hIcon			= LoadIcon(hInstance, (LPCTSTR)IDI_APPLICATION);
	wcex.hCursor		= LoadCursor(NULL, IDC_ARROW);
	wcex.hbrBackground	= (HBRUSH)(COLOR_WINDOW+1);
	wcex.lpszMenuName	= (LPCSTR)0;
	wcex.lpszClassName	= szWindowClass;
	wcex.hIconSm		= LoadIcon(wcex.hInstance, (LPCTSTR)IDI_APPLICATION);

	return RegisterClassEx( &wcex );
	}

BOOL InitInstance( HINSTANCE hInstance, int nCmdShow )
	{
	HWND hWnd;

	hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
		100, 100, 100, 100, NULL, NULL, hInstance, NULL);
	SetClientSize(hWnd,320,240);
	SetCenter(hWnd);

	if( !hWnd )
		return FALSE;

	ShowWindow( hWnd, nCmdShow );
	UpdateWindow( hWnd );

	return TRUE;
	}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
	{
	PAINTSTRUCT ps;
	HDC hdc;
	int i;

	switch( message )
		{
		case WM_PAINT:
			hdc = BeginPaint (hWnd, &ps);
			EndPaint( hWnd, &ps );
			break;
		case WM_KEYDOWN:
			if(wParam==VK_F1)
			joy->ChangeMode();
			mode=!mode;
				if(mode)
					SetWindowText(hWnd,"Color Arrow[joypad mode]");
				else
					SetWindowText(hWnd,"Color Arrow[keybord mode]");
			break;
		case WM_CREATE:
			CreateCDC(hWnd,dc,320,240);					//仮想画面作成
			col=PALETTERGB(255,255,255);
			Pen=CreatePen(PS_SOLID,2,col);
			oldp=(HPEN)SelectObject(dc,Pen);	//ペンを設定
			col=PALETTERGB(0,0,0);
			Brs=CreateSolidBrush(col);
			oldb=(HBRUSH)SelectObject(dc,Brs);			//ブラシを設定
			SetBkMode(dc,TRANSPARENT);					//文字の背景を
透明化
			SetTimer(hWnd,500,1,NULL);					//タイマー作成
			joy=new BCJOY(1);								//ク
ラスを作成
			srand((unsigned)time(NULL));				//乱数初期化
			for(i=0;i<5;i++)								//変
数初期化↓
				ht[i]=65535;
			scine=0;
			level=1;
			c=0;
			mode=0;
			break;
		case WM_DESTROY:
			delete joy;
			KillTimer(hWnd,500);
			SelectObject(dc,oldp);
```

★「エンターテイメント」「エンターテインメント」「エンタテイメント」「エンタテインメント」どれが正しいの？（埼玉県所沢市・藤 16歳）……【**影：**そんなこと，どうでも，いんたていめんと。**編：**電波新聞社では，エンターテインメントという表記をしています。】

```
				DeleteObject(Pen);
				SelectObject(dc,oldb);
				DeleteObject(Brs);
				DeleteDC(dc);
				PostQuitMessage( 0 );
				break;
			case WM_TIMER:
				switch(scine)
					{
					case 0:Scine0(hWnd);break;
					case 1:Scine1(hWnd);break;
					case 2:Scine2(hWnd);break;
					case 3:Scine3(hWnd);break;
					}
				break;
			default:
				return DefWindowProc( hWnd, message, wParam, lParam );
			}
		return 0;
		}

void Scine0(HWND hWnd)
		{
		HDC hdc;
		int i;
		char ti[]="COLOR ARROW";
		char ts[100];

		c++;
		level=(level+joy->dy()+3)%4 +1;

		hdc=GetDC(hWnd);
			PatBlt(dc,0,0,320,240,BLACKNESS);
			for(i=0;i<strlen(ti);i++)
				{
				if( (((int)(c/5)) % strlen(ti)) == i )
					SetTextColor(dc,RGB(255,0,10));
				else
					SetTextColor(dc,RGB(255,255,255));
				TextOut(dc,25+25*i+rand()%2,60+rand()%2,ti+i,1);
				}
			SetTextColor(dc,RGB(255,255,255));
			TextOut(dc,180,100,"Best Time",9);
			for(i=0;i<4;i++)
				{
				wsprintf(ts,"%5d",ht[i+1]);
				TextOut(dc,180,120+i*20,ts,5);
				}
			if(c>15)
				{
				TextOut(dc,60+rand()%2,120+rand()%2,"LEVEL 1",7);
				TextOut(dc,60+rand()%2,140+rand()%2,"LEVEL 2",7);
				TextOut(dc,60+rand()%2,160+rand()%2,"LEVEL 3",7);
				TextOut(dc,60+rand()%2,180+rand()%2,"LEVEL 4",7);
				TextOut(dc,30,100+level*20,">>",2);
				if(joy->buttonC()>0)
					scine=1;
				}
			BitBlt(hdc,0,0,320,240,dc,0,0,SRCCOPY);
		DeleteDC(hdc);

		return;
		}

void Scine1(HWND hWnd)
		{
		for(int i=0;i<50;i++)
			{
			bd[i]=rand()%4;
			bc[i]=rand()%level;
			}
		for(i=50;i<60;i++)
			bd[i]=5;
		c=0;
		t=0;
		scine=2;
		return;
		}


void Scine2(HWND hWnd)
		{
		HDC hdc;
		int i;
		int a[9]={0,0,2,1,0,2,1,3,3};
		bool h=false;
		COLORREF col[4]={RGB(100,255,100),RGB(255,100,100),RGB(150,150,255),RGB(255,255,100)};
		char ts[100];

		t++;

		if(joy->buttonC()==1 && joy->button(bc[c]+1) && a[joy->dir()-1]==bd[c])
			c++;

		hdc=GetDC(hWnd);
			PatBlt(dc,0,0,320,240,BLACKNESS);
			SetTextColor(dc,RGB(255,255,255));
			wsprintf(ts,"レベル:%d",level);
			TextOut(dc,10,10,ts,8);
			wsprintf(ts,"残り:%3d",50-c);
			TextOut(dc,100,10,ts,8);
			wsprintf(ts,"タイム:%5d",t);
			TextOut(dc,200,10,ts,12);
			Ellipse(dc,66,106,90,130);
			for(i=c;i<c+10;i++)
				{
				wsprintf(ts,"%s",(bd[i]==0)?"↓":(bd[i]==1)?"←":(bd[i]==2)?"→
":(bd[i]==3)?"↑":"  ");
				SetTextColor(dc,col[bc[i]]);
				TextOut(dc,71+(i-c)*20,111,ts,2);
				TextOut(dc,70+(i-c)*20,110,ts,2);
				}
			BitBlt(hdc,0,0,320,240,dc,0,0,SRCCOPY);
		DeleteDC(hdc);

		if(c==50)
			{
			c=0;
			scine=3;
			ht[level]=(ht[level]>t)?t:ht[level];
			}
		return;
		}
void Scine3(HWND hWnd)
		{
		HDC hdc;
		char ts[100];

		c++;

		hdc=GetDC(hWnd);
			PatBlt(dc,0,0,320,240,BLACKNESS);
			TextOut(dc,120,70,"Your Time",9);
			wsprintf(ts,"%5d",t);
			TextOut(dc,140,100,ts,5);
			if(ht[level]==t)
				TextOut(dc,100,130,"It's Best Time!!",16);
			BitBlt(hdc,0,0,320,240,dc,0,0,SRCCOPY);
		DeleteDC(hdc);

		if(c>15 && joy->buttonC()>0)
			{
			scine=0;
			c=0;
			}
		return;
		}
```

★突然ですが，みなさんワープロ・ソフトは，Word派それとも一太郎派？ちなみに僕はメモ帳派。(**神奈川県横浜市・バビブ改めボーちゃん20歳**)……【**編：**というわけで，今月の愛読者ハガキにワープロ・ソフトは何を使っているかのアンケートを入れてみました。】

Windows（C++Builder4）用



棒を使ってボールを突っつけ

BALL（ボール）てーじ

赤羊旅団（16歳）

ご挨拶

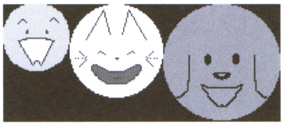
どうも，赤羊旅団の肉眼鏡Sです。

今回はジャンルの判別しがたい物を作ってみました。絵も内容も（ものすごく）雑ですが，息抜きぐらいにはなると思います。

内　容

このゲームは，自機である丸い顔を棒で突き，画面内を飛び交う青い玉にぶつけるゲームです。

青球にぶつける際には，当然跳ね返りますが，その衝撃で画面外に出るとゲーム・オーバー。うまく自機を突いてコントロールしてください。



《第1図》GMinuで使用する画像

操作法

クリック：棒で突く。自機の左のほうを突けば右へ，右のほうを突けば左へ飛びます。

また，なぜか空中にいる自機まで操れちゃいます。そうしないと，あっという間にゲーム・オーバーなので…。

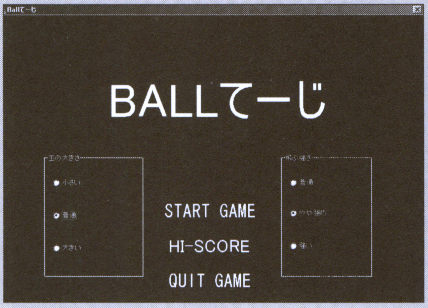
大きさなど

ゲームをスタートする前に，ラジオボタンで大きさや突く強さを選択できます。大きいと，青玉に当たりやすいのでハイスコアが狙えます。が，その分，跳ね返りが大きいので，上級者向けになっています。

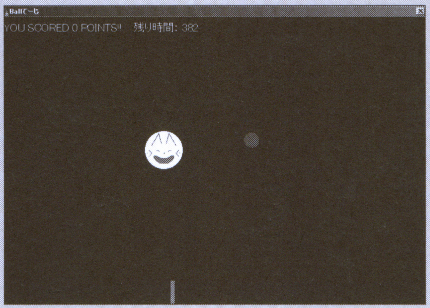
小さいほうは跳ね返りが少ないので，安定してゲームができます。

仕組み

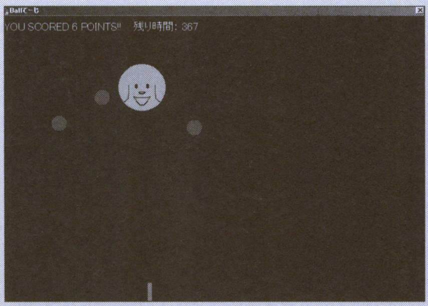
自機と青玉がぶつかったときは“|a||b| cos θ＝a*b”(aが自機と青球を結んだベクトル。bが青球の移動する方向のベクトル) とかでコサインを求め，“|b| cos θ”で力を分解し，自機の中心



《写真1》お好みの設定でゲーム・スタート



《写真2》顔のあるボールをうまく制御



《写真3》設定によってボールも変更できる

《写真4》ランキング機能も付いている

CHECKER FLAG

影：手軽に遊べるタイプのゲームだな。プレイ時間が短くなりやすいため，チョット物足りない感じもあるけど，しっかり作られていると言えるんじゃないかな？

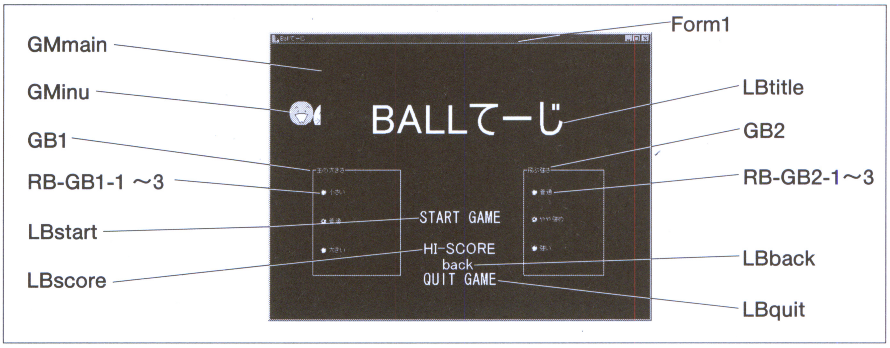
Dr.D：自機と青玉それぞれが，しっかりとした動きをしている部分に注目じゃな。リストでいうと，リスト2のmoveDrow( )関数じゃな。この関数で行っている処理が，このゲームの肝と言える。コメントがしっかりと記述されているので，どのような命令で移動処理を行っているかがわかるじゃろう。

《第1表》関数表

void __fastcall moveDrow( );	キャラクターを描画，移動させる
void __fastcall makeEnemy( );	敵機(青球)を生み出す
void __fastcall enemyKesu(int number);	割られた青球の処理
void __fastcall fileRead( );	セーブファイルを読み取る
void __fastcall fileWrite( );	セーブファイルを書き込む
void __fastcall obj(bool which);	コンポーネントの表示，非表示を一括切り替えする 引数のwhich＝true…表示，false…非表示

《第2表》変数表

click	クリックしたか記憶
Force	棒でつく力（ゲームスタート前に設定可能）
sj	0…セーブデータを読む 1…ゲームプレイ中 2…ハイスコアセーブ 3…タイトル画面へ移行する準備
mouseX	マウスのX座標。Yは必要ない
ballMAX	敵機（青球）の数
LV	timeによって増加する。青球の数の上限
now	タイマーに使用
zyunni	プレイヤーの順位を記憶 ハイスコア時の名前登録する時，デフォルトの名前にする
time	プレイタイム
name	名前を記憶
ziki	自機の構造体
.X, .Y	自機の座標
.r	自機の半径
.mx	絵のコピー開始場所
.moveX, .moveY	X，Y方向への移動量
.score	スコア
ball[20]	青球（敵機）の構造体
.X.Y	存在座標
.moveX.moveY	X，Y方向への移動量
HIscore[5]	ハイスコアの構造体
.score	スコア
.name	名前



《第2図》フォームの配置図

へかかる力を求めています。

本当は，さらに反発係数と運動量保存を使って跳ね返すのが本当らしいです。まだ授業でならっていないので，よくわからない…。

工夫？

1：ifを使いたくなかったので，こんなことをしてみました。“kaku＝atan（py/(px＋0.001)）;”これなら，PXが0のときでもエラーは出ませんよね？たぶん，0.001ぐらいなら計算結果にさほど影響は出ないと思いますし…。やっぱりifを使うほうがいいんでしょうか？

2：じつはこのゲーム，最初のころは，何もしないほうが高得点が出ていました。さすがにそれではまずいので，青球の跳ね上がる位置をボールより上にして，何もしないでいると得点が得られなくしてみました。

改造

・簡単なところで，自機の数を増やす。

・むずかしいところで，このシステム（棒で下から突いて自機操作）をアクションゲームにする。

・最初は敵を避けて進むが，一定時間もしくは特定のアイテムで体中に針が生える！それから一定時間は体当たりで敵を倒せる

……とか。

その他

団員madswireがHPを作りました。http://homepage2.nifty.com/madswire/です。日記は読み応えあります。ぜひ覗いてみてください。それと，現在PRGを作ってます。出来しだい投稿したいと思います。それでは，また。

《第3表》プロパティ表

フォームの設定	
Name	Form1
プロパティ	
AutoScroll	false
BorderIcons	[biSystemMenu]
BorderStyle	bsSingle
ClientHeight	571
ClinetWidth	792
Caption	Ballてーじ
Color	clBlack
Height	598
Left	213
Top	130
Width	800
イベント	
OnCreate	FormCreate
グループボックスの設定	
Name	GB1
Caption	玉の大きさ
Color	clBlack
Left	88
ParentFont	false
Top	256
Width	185
Name	GB2
Caption	飛ぶ強さ
Color	clBlack
Left	528
ParentFont	false
Top	256
Width	169
イメージの設定	
Name	GMinu
Height	49
Left	40
Picture	(TBitmap)
Top	120
Width	65
Name	GMmain
プロパティ	
Height	577
Left	0
Top	0
Width	793
イベント	
OnMouseDown	GMmainMouseDown
OnMouseMove	GMmainMouseMove
OnMouseUp	GMmainMouseUp
ラベルの設定	
Name	LBback
プロパティ	
Caption	back
Color	clBlack
Height	33
Left	360
ParentFont	false
Top	440
Visible	false
Width	64
イベント	
OnClick	LBbackClick
Name	LBquit
プロパティ	
Caption	QUIT GAME
Color	clBlack
Height	33
Left	320
ParentFont	false
Top	472
Width	153
イベント	
OnClick	LBquitClick
Name	LBscore
プロパティ	
AutoSize	false
Caption	HI-SCORE
Color	clBlack
Height	33
Left	320
ParentFont	false
Top	408
Width	153

★つい先日，NTTのPHSからfeel H”に変更。と思ったら，AirH”対応の新製品が発売！？ほしい！どうして，僕の物欲は，タイミングが悪いんだろう。PS2も値段が下がったし（PS2も2001年夏に買ったばかり）。**（東京都武蔵野市・風呂22歳）**……**【影：ぐはははは。】**

イベント	
OnClick	LBscoreClick
Name	LBstrat
プロパティ	
Caption	START GAME
Color	clBlack
Height	33
Left	312
ParentFont	false
Top	344
Visible	false
Width	170
イベント	
OnClick	LBstratClick
Name	LBtitle
Caption	BALLてーじ
Color clBlack	
Height	87
Left	208
ParentFont	false
Top	112
Width	406
ラジオボタンの設定	
Name	RB_GB1_1
プロパティ	
Caption	小さい
Color	clBlack
Height	25
Left	16
Top	40
Width	105
イベント	
OnClick	RB_GB1_1Clock
Name	RB_GB1_2
プロパティ	
Caption	普通
Color	clBlack
Height	17
Left	16
Top	104
Width	89
イベント	
OnClick	RB_GB1_2Clock
Name	RB_GB1_3
プロパティ	
Caption	大きい
Color	clBlack
Height	25
Left	16
Top	160
Width	97
イベント	
OnClick	RB_GB1_3Clock
Name	RB_GB2_1
プロパティ	
Caption	普通
Color	clBlack
Height	25
Left	16
Top	40
Width	121
イベント	
OnClick	RB_GB2_1Clock
Name	RB_GB2_2
プロパティ	
Caption	やや強め
Color	clBlack
Height	25
Left	16
Top	96
Width	113
イベント	
OnClick	RB_GB2_2Clock
Name	RB_GB2_3
プロパティ	
Caption	強い
Color	clBlack
Height	17
Left	16
Top	160
Width	113
イベント	
OnClick	RB_GB2_3Clock

《リスト1》「BALLてーじ」Balltaeg.cppのリスト

```
//---------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
USERES("Balltage.res");
USEFORM("Balltage1.cpp", Form1);
//---------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                Application->Initialize();
                Application->CreateForm(__classid(TForm1), &Form1);
                Application->Run();
        }
        catch (Exception &exception)
        {
                Application->ShowException(&exception);
        }
        return 0;
}
//--------------------
```

《リスト2》「BALLてーじ」Balltage1.cppのリスト

```
//---------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "math.h"
#include "Balltage1.h"
#include "stdlib.h"
#include "dos.h"
#include <stdio.h>
//--------------------
#include <dir.h>
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------
 void __fastcall TForm1::AppIdle(TObject *Sender,bool &Done)
{
 switch(sj){
  case 0:fileRead();sj=100;
  break;
  case 1:now=GetTickCount();        //現在の時間を得る
        moveDrow();makeEnemy();    //キャラを描いて、動かす；敵を作る
        while(now+29>GetTickCount())//29ミリ秒経っていなければ、待つ
        {now=now;}
  break;
  case 2:sj=3;fileWrite();        //ハイスコアを記憶
  break;
  case 3:sj=100;obj(true);LBback->Visible=false;  //必要なコンポーネントの表示、非表示
         GMmain->Canvas->Brush->Color=clBlack;
         GMmain->Canvas->Rectangle(-1,-1,800,800);//画面を消去
       LV=0;ziki.score=time=ballMAX=0;ziki.X=300;
         ziki.Y=ziki.moveX=ziki.moveY=0;//タイトルへ戻る準備
  break;
 }
 Done=false;
}
//---------------------------------------------------------------
void __fastcall TForm1::GMmainMouseDown(TObject *Sender,
      TMouseButton Button, TShiftState Shift, int X, int Y)
{
 if(click==0){
   //px…クリック地点と自機の中心との距離。r…半径
```

★ホームページでベーマガがまだ続いていると知り，1月号買いました。本屋でなかなか見つからなかったけど…。久々に見たベーマガ，15年くらい前と雰囲気が変わってなくて，昔を思い出してしまいました。(静岡市・いまはむかし35歳) ……**【編：また買って！】**

```
  int px=ziki.X-X,py;double bairitu,r=ziki.r;

  //a…半径は、px＊px＋py＊py＝r＊r なので、変形してpyを求める（このときは二乗した値）
  py=r*r-px*px;

  if(py<0){
    return;//ゼロより小さい時は棒が自機にあたっていない。
  }
  py=-sqrt(py);bairitu=Force/r;
  //pyは二乗した値であった為、平方を取る；倍率が半径に左右されないように倍率を決める

  ziki.moveX+=px*bairitu;     //x方向に加える
  ziki.moveY*=0.5;
  ziki.moveY+=py*bairitu;     //y方向に加える
 }
 click=1;
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－
void __fastcall TForm1::LBstratClick(TObject *Sender)//スタートを押した
{
 obj(false);//コンポーネントを隠す；
 sj=1;GMmain->Canvas->Brush->Color=clBlack;
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－
//半径、跳ね返る時の強さ、絵の開始位置設定
void __fastcall TForm1::RB_GB1_1Click(TObject *Sender)
{
  ziki.r=25;ziki.hannsha=-0.3;ziki.mx=0;
}
void __fastcall TForm1::RB_GB1_2Click(TObject *Sender)
{
  ziki.r=35;ziki.hannsha=-0.2;ziki.mx=50;
}
void __fastcall TForm1::RB_GB1_3Click(TObject *Sender)
{
 ziki.r=44;ziki.hannsha=0;ziki.mx=120;
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－
void __fastcall TForm1::RB_GB2_1Click(TObject *Sender)//力の設定
{
 Force=20;//力、普通
}
void __fastcall TForm1::RB_GB2_2Click(TObject *Sender)
{
  Force=24;//力、やや強い
}
void __fastcall TForm1::RB_GB2_3Click(TObject *Sender)
{
  Force=27;//力、強い
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－

void __fastcall TForm1::FormCreate(TObject *Sender)
{
   ziki.X=320;ziki.Y=70;ziki.r=35;ziki.mx=50;Force=24;ziki.hannsha=-0.1;
   GMmain->Canvas->Font->Color=clWhite;
```

```
   GMmain->Canvas->Font->Size=15;
   GMmain->Canvas->Brush->Color=clBlack;
   GMmain->Canvas->Rectangle(-1,-1,800,600);
   Application->OnIdle=AppIdle;
   randomize();
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－

void __fastcall TForm1::GMmainMouseUp(TObject *Sender, TMouseButton Button,
      TShiftState Shift, int X, int Y)
{
  click=0;
}
//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－
void __fastcall TForm1::moveDrow()
{
 GMmain->Canvas->Rectangle(0,0,793,577);  //全体を消す

 ziki.X+=ziki.moveX;ziki.Y+=ziki.moveY;   //動かす
 int r=ziki.r,Py=450+r;                   //py…地面の高さ
 ziki.moveY++;                            //重力加速度を足す（1です）
 if(ziki.X<0 || ziki.X>793 || ziki.Y>650){sj=2;  //画面からはみ出ているか？
   GMmain->Canvas->Font->Color=clWhite;
   GMmain->Canvas->TextOutA(300,240,"ゲームオーバー！");
   String str="得点： ";str+=ziki.score;
   GMmain->Canvas->TextOutA(330,270,str);
   GMmain->Refresh();Sleep(1500);
 }
 if(ziki.Y>450 ){ziki.moveY*=-0.55;ziki.moveX*=0.35;ziki.Y=450;}//地面にぶつかっているか？
 if(ziki.Y<r){ziki.moveY*=-0.55;ziki.moveX*0.35;ziki.Y=r;}       //天井にぶつかっているか？
 BitBlt(GMmain->Canvas->Handle,ziki.X-r,ziki.Y-r,2*r,2*r,GMinu->Canvas-<
>Handle,ziki.mx,0,cmSrcCopy);//絵を貼り付け

 GMmain->Canvas->Brush->Color=clRed;
 GMmain->Canvas->Rectangle(mouseX-5,Py-click*50,mouseX+5,Py-click*50+120);//棒を描く

 GMmain->Canvas->MoveTo(0,Py);GMmain->Canvas->LineTo(792,Py);  //ステージを表す線を描く

 GMmain->Canvas->Brush->Color=clBlue;
 for(int k=0;k<ballMAX;k++){                              //ボールの数だけ繰り返す
   ball[k].X+=ball[k].moveX;ball[k].Y+=ball[k].moveY;  //ボールを動かす
   ball[k].moveY++;                                       //重力加速度を足す
   //玉を表す円を描く
   GMmain->Canvas->Ellipse(ball[k].X-15,ball[k].Y-15,ball[k].X+15,ball[k].Y+15);
   if(ball[k].Y>380){ball[k].moveY*=-1;ball[k].Y=380;}//地面にぶつかった
   if(ball[k].X<0 || ball[k].X>793){                     //画面外へ出た
     enemyKesu(k);
   }
   double px=ziki.X-ball[k].X,py=ziki.Y-ball[k].Y; //pxは自機と玉[k]の水平距離。pyは垂直距離
   double kyori=sqrt(px*px+py*py+0.001),kaku,kyoriBallMove;//三平方の定理より距離を求める
   if(kyori<r+15){                                          //自機とぶつかった
        kyoriBallMove=sqrt(ball[k].moveX*ball[k].moveX+ball[k].moveY*ball[k].moveY);
     kaku=(ball[k].moveX*px+ball[k].moveY*py)/(kyori*kyoriBallMove);
     //↑aベクトル、bベクトルの角はcosθ=(ax*bx+ay*by)/(|a|*|b|)より、
     //自機と玉を結ぶ直線と玉の移動方向との、角のコサインを得る。
```


★「センター試験対策」ありがとうございました。なんだか，20点もらった気分です。次号も楽しみにしています。（千葉県浦安市・初男17歳）……**【影：**今月も読んでくれていることでしょう。これで，40点になっちゃいましたね。もう安心？！**編：**試験がんばってね。**】**

```
      kyoriBallMove*=kaku;        //上で求めた事から、自機の中心への力を求める。
      kaku=atan(py/(px+0.0001));  //自機と玉の角度を求める。
      if(px<0){kaku+=3.14;}
      ziki.moveX+=kyoriBallMove*cos(kaku)*(1+ziki.hannsha);//X方向へ力を足す
      ziki.moveY+=kyoriBallMove*sin(kaku)*(1.2+ziki.hannsha);//Y方向へ力を足す
      enemyKesu(k);ziki.score+=3;
    }
  }
  GMmain->Canvas->Brush->Color=clBlack;
  String str="YOU SCORED ";str+=ziki.score;str+=" POINTS!!"; //得点
  GMmain->Canvas->TextOutA(10,10,str);
  str="残り時間：  ";str+=int(400-time);                    //残り時間
  GMmain->Canvas->TextOutA(250,10,str);
  time+=0.5;                    //時間を足す
  if(time>LV*40+40){LV++;       //LVアップ（玉の数が増える）
    if(LV==10){                 //タイムアップ
    sj=2;
    GMmain->Canvas->TextOutA(320,240,"時間切れ！");  //得点表示
    str="得点：  ";str+=ziki.score;
    GMmain->Canvas->TextOutA(340,270,str);
    GMmain->Refresh();Sleep(1500);
    }
  }
}
//-------------------------------------------------------
void __fastcall TForm1::makeEnemy()
{
  if(random(300-LV*10)==5 || ballMAX<LV*0.5+1){  //玉を作るか？
  int rnd=random(2);                              //画面の左から出るか、右から出るか
   ball[ballMAX].X=rnd*792;ball[ballMAX].Y=random(350)+50;
   ball[ballMAX].moveX=-(rnd*2-1)*(random(15)+7);
   ball[ballMAX].moveY=0;                   //座標、スピードを決定
   ballMAX++;
  }
}
void __fastcall TForm1::enemyKesu(int k)
{
   ballMAX--;
   ball[k].X=ball[ballMAX].X;ball[k].Y=ball[ballMAX].Y;
   ball[k].moveX=ball[ballMAX].moveX;ball[k].moveY=ball[ballMAX].moveY;
}                         //↑消えた玉の番号[k]に一番後に作られた玉のデータをいれ、
                          //  玉の数を減らす
void __fastcall TForm1::GMmainMouseMove(TObject *Sender, TShiftState Shift,
      int X, int Y)
{
 mouseX=X;                //マウスのX座標を得る
}
//-------------------------------------------------------
void __fastcall TForm1::fileRead()
{ int hFile,len;
  char buff[10000];
  name="HI-SCORE.dat";          //ファイルネーム
  if(!FileExists(name)){        //ファイルがあるか？
     hFile=FileCreate(name);
     for(int k=0;k<5;k++){      //ないときは、作って、ハイスコアを「名前無し」に
        HIscore[k].name="名前なし";HIscore[k].score=0;
     }
     FileClose(hFile);
     return;
  }
  else{
    hFile=FileOpen(name,fmOpenRead);      //あるときは読み取る。
      if(hFile!=-1){
        len=FileRead(hFile,buff,10000);
        FileClose(hFile);
        if(buff[0]=='&'){       //書き込まれているか？
          String str;int a=1;       //書き込まれているので、読み出す。
          for(int k=0;k<5;k++){
            while(buff[a]!=':'){HIscore[k].name+=buff[a];a++;}a++;
            while(buff[a]!=':'){str+=buff[a];a++;}a++;
            HIscore[k].score=str.ToInt(); str="";
          }
        }
      }
  }
}
void __fastcall TForm1::fileWrite()
{
 for(int k=0;k<5;k++){
    if(HIscore[k].score<ziki.score){//今の点数はハイスコアか？
      int c=3;
      while(c>=k)//プレイヤーより下の順位を、一位ずつずらす
      {HIscore[c+1].name=HIscore[c].name;HIscore[c+1].score=HIscore[c].score;c--;}
      HIscore[k].score=ziki.score;
      name="おめでとう！";name+=k+1;name+="位入賞です";
      AnsiString nm=InputBox("名前入力",name,HIscore[zyunni].name);//名前入力
      HIscore[k].name=nm;
      zyunni=k;//とった順位記憶（名前入力のときに使う）
      k=5;
    }
 }
 String buff='&',str;int a=1;//aは、ファイルの大きさ
 for(int k=0;k<5;k++){
      str=HIscore[k].score;
      a+=HIscore[k].name.Length()+str.Length()+2;
      buff+=HIscore[k].name;buff+=':';     //' : ' でくぎる
      buff+=HIscore[k].score;buff+=':';    //同上
 }
 int hFile=FileOpen("HI-SCORE.dat",fmOpenWrite);
 int len=FileWrite(hFile,buff.c_str(),a);
 if(len==-1){
         Beep();MessageDlg("ファイル書き込みエラー",mtWarning,TMsgDlgButtons()<<mbOK,0);}
 FileClose(hFile);
}
//-------------------------------------------------------
void __fastcall TForm1::LBscoreClick(TObject *Sender)
{
 obj(false);LBscore->Visible=true;
 for(int k=0;k<64;k++){
   LBscore->Top-=5;//上にずらす
   Form1->Refresh();Sleep(2);
 }
```

★ゲームのことはやらないのですか？（愛知県名古屋市・**陽炎14歳**）……【**編：**プログラミングを中心に，コンピュータのしくみや原理が学べる内容を掲載していきます。ゲームがどんな言語で書かれていて，どんなふうに出来上がるのかという内容はアリですね。】

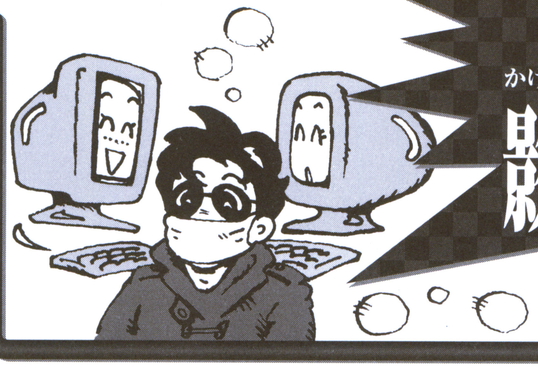
```
 GMmain->Visible=true;
 GMmain->Canvas->Font->Color=clWhite;
 for(int k=0;k<5;k++){
   name=HIscore[k].name;name+="  score = ";name+=HIscore[k].score;
   GMmain->Canvas->TextOutA(245,k*55+150,name);
 }
 LBback->Visible=true;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::obj(bool which)//ラベルなどを一括して表示、非表示を切り替える
{
  GB1->Visible=which;GB2->Visible=which;LBstrat->Visible=which;LBtitle->Visible=which;
 LBquit->Visible=which;LBscore->Visible=which;LBback->Visible=which;
}
void __fastcall TForm1::LBbackClick(TObject *Sender)//タイトル画面に戻す
{
 obj(true);
 LBscore->Top+=320;
 LBback->Visible=false;
 GMmain->Canvas->Rectangle(-1,-1,801,601);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::LBquitClick(TObject *Sender)
{
   Close();//閉じる
}
//---------------------------------------------------------------------------
```

《リスト3》「BALLてーじ」Balltage1.hのリスト

```
//---------------------------------------------------------------------------
#ifndef Unit1H
#define Unit1H
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//---------------------------------------------------------------------------
class TForm1 : public TForm
{
__published:        // IDE 管理のコンポーネント
        TImage *GMmain; TLabel *LBstrat; TLabel *LBquit; TLabel *LBtitle;
        TGroupBox *GB1;        TGroupBox *GB2;
        TRadioButton *RB_GB1_2;     TRadioButton *RB_GB1_3;
        TRadioButton *RB_GB1_1;     TRadioButton *RB_GB2_3;
        TRadioButton *RadioButton2;TRadioButton *RB_GB2_2;
        TRadioButton *RB_GB2_1;
        TLabel *LBscore;        TLabel *LBback;
        TImage *GMinu;
        void __fastcall GMmainMouseDown(TObject *Sender,
          TMouseButton Button, TShiftState Shift, int X, int Y);
        void __fastcall LBstratClick(TObject *Sender);
        void __fastcall RB_GB1_1Click(TObject *Sender);
        void __fastcall RB_GB1_2Click(TObject *Sender);
        void __fastcall RB_GB1_3Click(TObject *Sender);
        void __fastcall RB_GB2_1Click(TObject *Sender);
        void __fastcall RB_GB2_2Click(TObject *Sender);
        void __fastcall RB_GB2_3Click(TObject *Sender);
        void __fastcall FormCreate(TObject *Sender);
        void __fastcall GMmainMouseUp(TObject *Sender, TMouseButton Button,
          TShiftState Shift, int X, int Y);
        void __fastcall GMmainMouseMove(TObject *Sender, TShiftState Shift,
          int X, int Y);
        void __fastcall LBscoreClick(TObject *Sender);
        void __fastcall LBbackClick(TObject *Sender);
        void __fastcall LBquitClick(TObject *Sender);
private:  // ユーザー宣言
        void __fastcall AppIdle(TObject *Sender,bool &Done);
        void __fastcall moveDrow();void __fastcall makeEnemy();
        void __fastcall enemyKesu(int number);
        void __fastcall fileRead();void __fastcall fileWrite();
        void __fastcall obj(bool which);

        int click,Force,sj,mouseX,ballMAX,LV,now,zyunni;
        double time;
        String name;
struct ZIKI{
int X,Y,moveX,moveY,r,score,mx;
double hannsha;
};ZIKI ziki;
struct BALL{
int X,Y,moveX,moveY;

};BALL ball[20];
struct HISCORE{
int score;
String name;
};HISCORE HIscore[5];




public:             // ユーザー宣言
        __fastcall TForm1(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;
//---------------------------------------------------------------------------
#endif
```


★今月号（2002年1月号）の表紙はなんとなくお正月っぽいと思ったのは私だけ？（北海道旭川市・**わかまつ16歳**）……**【つぐ美：**他にもこんなようなコメント書いてくれた人いたわよ。ウイルスバスター2002の赤いパッケージのおかげで，PCがお餅のようだとか。**】**

Windows（F-BASIC V6.3）用



影さんが登場する伝統ゲーム

影（かげ）さんのパソ破（は）ッ！（十六武蔵）

ロディマス（29歳）

ストーリー

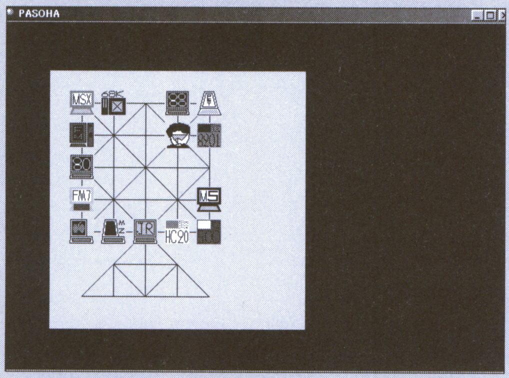
「あーあ，もうこんな古いパソコンいらないよなあ」

影さんのパソ破！が始まった…。

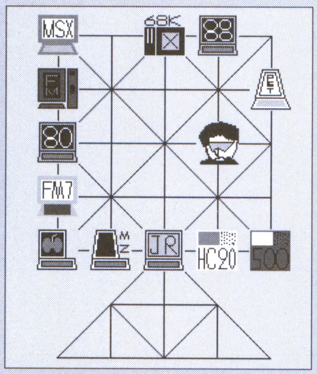
内　容

俗に言う十六武蔵です。影さんが親で，パソコンとパソコンの間に入ると，両側のパソコンを破壊できます。影さんが最初に動くので，最初の斜めのムーブで，パソコンは無条件に二つ「パソ破」されます。

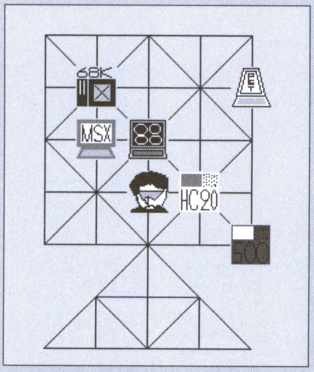
あなたは各パソコンを操り，影さんを動けなくしたら勝ちです。パソコンの数が二つになったら，あなたの負けです。コンピュータの思考ルーチンはなかなか強いと自負しています。



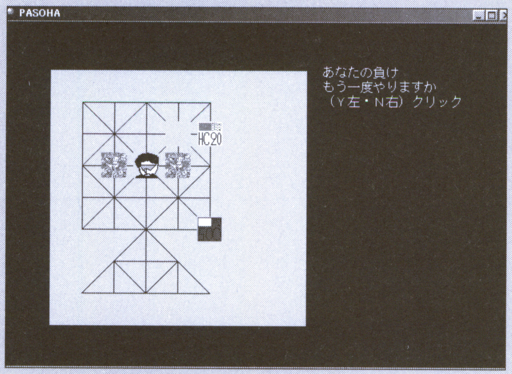
《写真1》古いパソコンで影さんを追い込め



《写真2》うまく誘導していくのがコツかも？



《写真3》だが，このゲームの影さんはかなり手強い



《写真4》慣れていないとすぐにゲーム・オーバーに

《第1表》画像ファイル表

画像	ファイル名	画像	ファイル名
60	60.bmp	M5	M5.bmp
68K	68k.bmp	MSX	MSX.bmp
80	80.bmp	MZ	Mz.bmp
8201	8201.bmp	PET	Pet.bmp
88	88.bmp	FM	Towns.bmp
	Apple.bmp	X1	X1.bmp
500	E500.bmp		Kage.bmp
FM7	Fm7.bmp		Hakai.bmp
HC20	Hc20.bmp		Blank.bmp
JR	Jr.bmp		

※サイズはすべて32×32

★家族の趣味はすべて頭にパがつきます。私がパソコンで，兄はパラグライダー，姉はパッチワーク，父はパチンコ，母は…パパ（父）です。（東京都世田谷区・パパパパパフィー17歳）……【影：編くんがパズルで，つぐ美ちゃんがパラパラ。私がパ・パ・パ…うーん。】

影さんのパソ破ッ！（十六武蔵）のプログラム・リスト

```
#include "windows.bi"
randomize TIMER
declare sub DISPLAY()
declare sub COMPUTER()
declare sub MAN()
declare function MOUSEWAIT()
declare function MAN_INPUT_CHECK(IN_OR_OUT%)
declare function PASOHA(ENEMY1%,ENEMY2%)
declare function NANAME_CHECK(KOMA%)
declare function NEW_CHECK(OLD%,CLIST%)
declare function COM_MOVE_CHECK(OLD%,CLIST%)
declare function MAN_MOVE_CHECK(OLD%,NEW%)
var shared BOARD(16*16+2) as double
var shared VALUE(16*16+2) as double
var shared WIN_LOSE_CHECK,MOUSECHECK as integer
var shared BLANK as object
var shared KAGE as object
var shared HAKAI as object
var shared KOMA(16) as object
BITMAPOBJECT BLANK:BITMAPOBJECT KAGE:BITMAPOBJECT HAKAI
for K%=0 to 16:BITMAPOBJECT KOMA(K%):next K%
BLANK.LOADFILE "BLANK.BMP"    :KAGE.LOADFILE "kage.BMP"     :KOMA(1).LOADFILE "msx.BMP"
KOMA(2).LOADFILE "68k.BMP"    :KOMA(3).LOADFILE "60.BMP"    :KOMA(4).LOADFILE "88.BMP"
KOMA(5).LOADFILE "pet.BMP"    :KOMA(6).LOADFILE "apple.BMP" :KOMA(7).LOADFILE "mz.BMP"
KOMA(8).LOADFILE "jr.BMP"     :KOMA(9).LOADFILE "hc20.BMP"  :KOMA(10).LOADFILE "e500.BMP"
KOMA(11).LOADFILE "towns.BMP" :KOMA(12).LOADFILE "80.BMP"   :KOMA(13).LOADFILE "fm7.BMP"
KOMA(14).LOADFILE "8201.BMP"  :KOMA(15).LOADFILE "x1.BMP"   :KOMA(16).LOADFILE "m5.BMP"
HAKAI.LOADFILE "hakai.bmp"
cls
'盤の周囲を壁で囲む
for I%=0 to 256:BOARD(I%)=99:VALUE(I%)=99:next I%
while 1
        cls
        '盤の中を空白にする
        for X% = 2 to 6: for Y% = 2 to 8:BOARD(X%*16+Y%) = 0:next Y%:next X%
        BOARD(2*16+7)=99:BOARD(6*16+7)=99
        '駒を置く
        BOARD(4*16+4) =20:BOARD(2*16+2) = 1:BOARD(3*16+2) = 2:BOARD(4*16+2) = 3
        BOARD(5*16+2) = 4:BOARD(6*16+2) = 5:BOARD(2*16+6) = 6:BOARD(3*16+6) = 7
        BOARD(4*16+6) = 8:BOARD(5*16+6) = 9:BOARD(6*16+6) =10:BOARD(2*16+3) =11
        BOARD(2*16+4) =12:BOARD(2*16+5) =13:BOARD(6*16+3) =14:BOARD(6*16+4) =15
        BOARD(6*16+5) =16
        '各マスに点数をつける
        VALUE(2*16+8)=-10:VALUE(3*16+8) =-10:VALUE(4*16+8)=-10 :VALUE(5*16+8)=-10
        VALUE(6*16+8)=-10:VALUE(3*16+7) = -9:VALUE(4*16+7)= -9 :VALUE(5*16+7)= -9
        VALUE(2*16+6)= -9:VALUE(3*16+6) = -8:VALUE(4*16+6)= -8 :VALUE(5*16+6)= -8
        VALUE(6*16+6)= -9:VALUE(2*16+2) = -9:VALUE(3*16+2)= -8 :VALUE(4*16+2)= -7
        VALUE(5*16+2)= -8:VALUE(6*16+2) = -9:VALUE(6*16+3)= -8 :VALUE(6*16+4)= -7
        VALUE(6*16+5)= -8:VALUE(2*16+3) = -8:VALUE(2*16+4)= -7 :VALUE(2*16+5)= -8
        VALUE(3*16+3)=  0:VALUE(3*16+4) =  0:VALUE(3*16+5)=  0 :VALUE(4*16+3)=  0
        VALUE(4*16+4)= 10:VALUE(4*16+5) =  0:VALUE(5*16+3)=  0 :VALUE(5*16+4)=  0
        VALUE(5*16+5)=  0
        DISPLAY:WIN_LOSE_CHECK=0
        while 1
                COMPUTER:if WIN_LOSE_CHECK<>0 then goto *ENDING
                wait 10:DISPLAY:MAN:DISPLAY
        wend
        *ENDING
        if WIN_LOSE_CHECK=1 then locate 50,3:print "私の負けです"
        if WIN_LOSE_CHECK=2 then locate 50,3:print "あなたの負け"
        locate 50,4:print "もう一度やりますか"
        locate 50,5:print " (Y左・N右) クリック"
        MOUSECHECK=mousewait:if MOUSECHECK=1 then goto *JUMP
wend
*JUMP
BLANK.DELETEOBJECT:KAGE.DELETEOBJECT:for K%=1 to 16:KOMA(K%).DELETEOBJECT:next K%
HAKAI.DELETEOBJECT:end
sub DISPLAY()'盤と駒を描く
var I%,J%
        '盤を描く
        line(40*1+20,40*1+20)-(40*9+20,40*9+20),pset,9,bf
        for I%=2 to 6:line(40* 2+20,40*I%+20)-(40* 6+20,40*I%+20),pset,0:next I%
        line(40*3+20,40*7+20)-(40*5+20,40*7+20),pset,0
        line(40*2+20,40*8+20)-(40*6+20,40*8+20),pset,0
        for I%=2 to 6:line(40*I%+20,40* 2+20)-(40*I%+20,40* 6+20),pset,0:next I%
        line(40*4+20,40*6+20)-(40*4+20,40*8+20),pset,0
        line(40*2+20,40*4+20)-(40*6+20,40*8+20),pset,0
        line(40*2+20,40*2+20)-(40*6+20,40*6+20),pset,0
        line(40*6+20,40*6+20)-(40*2+20,40*2+20),pset,0
        line(40*4+20,40*2+20)-(40*6+20,40*4+20),pset,0
        line(40*6+20,40*2+20)-(40*2+20,40*6+20),pset,0
        line(40*2+20,40*8+20)-(40*6+20,40*4+20),pset,0
        line(40*2+20,40*4+20)-(40*4+20,40*2+20),pset,0
        line(40*3+20,40*7+20)-(40*3+20,40*8+20),pset,0
        line(40*5+20,40*7+20)-(40*5+20,40*8+20),pset,0
        line(40*3+20,40*7+20)-(40*4+20,40*8+20),pset,0
        line(40*5+20,40*7+20)-(40*4+20,40*8+20),pset,0
        '駒を描く
        for I%=2 to 6:for J%=2 to 8
                if BOARD(I%*16+J%)= 99 then DRAWBITMAP BLANK,40*I%+4,40*J%+4
                if BOARD(I%*16+J%)= 20 then DRAWBITMAP KAGE,40*I%+4,40*J%+4
                for K%=1 to 16
                        if BOARD(I%*16+J%)= K% then DRAWBITMAP KOMA(K%),40*I%+4,40*J%+4
                next K%
        next J%:next I%
end sub
sub COMPUTER()'コンピュータの思考ルーチン
var OLD%,NEW%,I%,CLIST%,ENEMY1%,ENEMY2%,ENEMY_CHECK%,NNEW%(20),EV2%,EV1%
var MOVE_CHECK%
        '狐の位置を探し出しＯＬＤ%に保存する
        for I%=0 to 256:if BOARD(I%)=20 then OLD%=I%
        next I%
        MOVE_CHECK%=0:CLIST%=0:SP%=0:EV2%=-30000
        while 1:EV1%=0:CLIST%=COM_MOVE_CHECK(OLD%,CLIST%)
                if CLIST%>10 then goto *JUMP
                I%=NEW_CHECK(OLD%,CLIST%):EV1%=EV1%+VALUE(I%):ENEMY1%=I%+ 1:ENEMY2%=I%- 1
                if BOARD(ENEMY1%)>0 and BOARD(ENEMY1%)<17 and BOARD(ENEMY2%)>0 and
BOARD(ENEMY2%)<17 then EV1%=EV1%+100
                ENEMY1%=i%+16:ENEMY2%=i%-16
                if BOARD(ENEMY1%)>0 and BOARD(ENEMY1%)<17 and BOARD(ENEMY2%)>0 and
BOARD(ENEMY2%)<17 then EV1%=EV1%+100
                '或る場所にある駒は斜めの敵を獲れません
                CHECK%=NANAME_CHECK(I%):if CHECK%=0 then goto *SELECT
                ENEMY1%=I%+17:ENEMY2%=I%-17
                if BOARD(ENEMY1%)>0 and BOARD(ENEMY1%)<17 and BOARD(ENEMY2%)>0 and
BOARD(ENEMY2%)<17 then EV1%=EV1%+100
                ENEMY1%=i%+15:ENEMY2%=i%-15
                if BOARD(ENEMY1%)>0 and BOARD(ENEMY1%)<17 and BOARD(ENEMY2%)>0 and
BOARD(ENEMY2%)<17 then EV1%=EV1%+100
*SELECT
                CLIST%=CLIST%+1:if EV1%<EV2% then goto *JUMP2
```

CHECKER FLAG

つぐ美：ロディマスさんお得意のCOM思考ルーチン重視のテーブル・ゲームね。ゲーム自体がチョット頭を使う形式なのに加えて，このCOMの強さですから，はじめてこのゲームをプレイする人には，ある意味オススメできません。あと，チョット残念なのが，多くの画像ファイルを用意しないと遊べない点ですね。

Dr.D：そのCOMルーチン部分の処理を見ると，細かな条件文を指定したif文がいくつも使われていたりなど，かなり凝ったルーチンであることがパッと見でわかるじゃろう。だが，COMは常に同じ思考でコマを動かしていくので，何回かプレイするとCOMのクセが読めてくるゾ。

影：フッフッフッ，私のパソ破は誰にも止められないのだよ，編クン。

編：うっ，確かにこのゲーム内では，そう言われても仕方ないかもしれません…。

★参考にしたもの

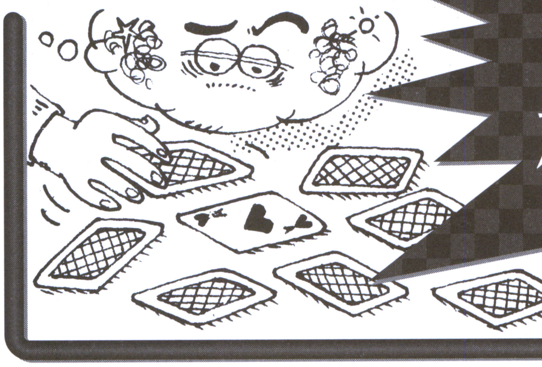
・「世界のゲーム事典」松田道弘著／東京堂出版

★バザーでMDウォークマン手に入れました。500円。ただし録音機能がついているのに使えない。再生だけあればいいんだけど，どうも気になる。（静岡県浜北市・**小島亮一20歳**）……【**編：**地域のバザーって，掘り出し物があるんだよね。**影：**五反田近辺はどこどこ？】

```
                    if EV1%>EV2% then SP%=0:NNEW%(SP%)=I%:EV2%=EV1%:MOVE_CHECK%=1:goto *JUMP2
                    SP%=SP%+1:NNEW%(SP%)=I%
*JUMP2
          wend
*JUMP
          if MOVE_CHECK%=0 then goto *ENDING
          SP%=cint(rnd(1)*SP%):NEW%=NNEW%(SP%):CLIST%=0:BOARD(OLD%)=0:BOARD(NEW%)=20
          ENEMY1%=NEW%+ 1:ENEMY2%=NEW%- 1:ENEMY_CHECK%=PASOHA(ENEMY1%,ENEMY2%)
          ENEMY1%=NEW%+16:ENEMY2%=NEW%-16:ENEMY_CHECK%=PASOHA(ENEMY1%,ENEMY2%)
          '或る場所にある駒は斜めの敵を獲れません
          CHECK%=NANAME_CHECK(NEW%):if CHECK%=0 then goto *SELECT4
          ENEMY1%=NEW%+17:ENEMY2%=NEW%-17:ENEMY_CHECK%=PASOHA(ENEMY1%,ENEMY2%)
          ENEMY1%=NEW%+15:ENEMY2%=NEW%-15:ENEMY_CHECK%=PASOHA(ENEMY1%,ENEMY2%)
*SELECT4
          I%=(NEW%  ¥ 16):J%=(NEW%  mod 16):DRAWBITMAP KAGE,40*I%+4,40*J%+4
          I%=(OLD%  ¥ 16):J%=(OLD%  mod 16):DRAWBITMAP BLANK,40*I%+4,40*J%+4
*ENDING
          '狐が動けなくなったらコンピュータの負け
          if CLIST%>10 then WIN_LOSE_CHECK=1
          'ガチョウの数が２個になったらプレイヤーの負け
          GEESE%=0:for I%=0 to 256:if BOARD(I%)>0 and BOARD(I%)<17 then GEESE%=GEESE%+1
          next I%:if GEESE%<=2 then WIN_LOSE_CHECK=2
end sub
function PASOHA(ENEMY1%,ENEMY2%)
var I%,J%
          PASOHA=0
          if ENEMY1%<0 or ENEMY1%>256 or ENEMY2%<0 or ENEMY2%>256 then goto *ENDING
          if (BOARD(ENEMY1%)<1 or BOARD(ENEMY1%)>16) then PASOHA=1:goto *ENDING
          if (BOARD(ENEMY2%)<1 or BOARD(ENEMY2%)>16) then PASOHA=1:goto *ENDING
          BOARD(ENEMY1%)=0:BOARD(ENEMY2%)=0:I%=(ENEMY1% ¥ 16):J%=(ENEMY1% mod 16)
          DRAWBITMAP HAKAI,40*I%+4,40*J%+4 :I%=(ENEMY2% ¥ 16):J%=(ENEMY2% mod 16)
          DRAWBITMAP HAKAI,40*I%+4,40*J%+4 :beep
*ENDING
end function
sub MAN()
var X%,Y%,OLD%,NEW%,CHECK%,MOVE_CHECK%,KOMA%
          MOVE_CHECK%=0
          while MOVE_CHECK%=0
                    OLD%=MAN_INPUT_CHECK(1):KOMA%=BOARD(OLD%):NEW%=MAN_INPUT_CHECK(2)
                    MOVE_CHECK%=MAN_MOVE_CHECK(OLD%,NEW%)
                    if MOVE_CHECK%=0 then locate 10,1:print "そこには進めません。":X%=(OLD% ¥
16):Y%=(OLD% mod 16):line(X%*40+3,Y%*40+3)-step(32,32),pset,9,b
          wend:BOARD(OLD%)=0:BOARD(NEW%)=KOMA%
end sub
function MAN_INPUT_CHECK(IN_OR_OUT%)
var CHECK%,CELL%
          while CHECK%=0
                    MOUSECHECK=mousewait:X%=int(mouse(0)/40):Y%=int(mouse(1)/40)
                    CELL%=X%*16+Y%:locate 10,1:print "                    "
                    if CELL%<0 and CELL%>255 then goto *ENDING
                    if IN_OR_OUT%=2 then CHECK%=1:goto *ENDING
                    if BOARD(CELL%)>0 and BOARD(CELL%)<17 then CHECK%=1:line(X%*40+3,Y%*40+3)-
step(32,32),pset,5,b else locate 10,1:print "其処に駒はありません"
*ENDING
          wend:MAN_INPUT_CHECK=CELL%
end function
function NEW_CHECK(OLD%,CLIST%)
          if CLIST%=0 then NEW_CHECK=OLD%+16
          if CLIST%=1 then NEW_CHECK=OLD%-16
          if CLIST%=2 then NEW_CHECK=OLD%-1
          if CLIST%=3 then NEW_CHECK=OLD%+1
          if CLIST%=4 then NEW_CHECK=OLD%+15
          if CLIST%=5 then NEW_CHECK=OLD%-15
          if CLIST%=6 then NEW_CHECK=OLD%+17
          if CLIST%=7 then NEW_CHECK=OLD%-17
end function
function COM_MOVE_CHECK(OLD%,CLIST%)
          if CLIST%=0 then goto *LIST0
          if CLIST%=1 then goto *LIST1
          if CLIST%=2 then goto *LIST2
          if CLIST%=3 then goto *LIST3
          if CLIST%=4 then goto *LIST4
          if CLIST%=5 then goto *LIST5
          if CLIST%=6 then goto *LIST6
          if CLIST%=7 then goto *LIST7
          CLIST%=100:goto *ENDING
*LIST0:   CLIST%=0:NEW%=OLD%+16:if BOARD(NEW%)<>0 then goto *LIST1
          goto *ENDING
*LIST1:   CLIST%=1:NEW%=OLD%-16:if BOARD(NEW%)<>0 then goto *LIST2
          goto *ENDING
*LIST2:   CLIST%=2:NEW%=OLD%- 1:if OLD%=87 and NEW%=86 then goto *LIST3
          if OLD%=55 and NEW%=54 then goto *LIST3
          if BOARD(NEW%)<>0 then goto *LIST3
          goto *ENDING
*LIST3:   CLIST%=3
          NEW%=OLD%+ 1:if OLD%=86 and NEW%=87 then goto *LIST4
          if OLD%=54 and NEW%=55 then goto *LIST4
          if BOARD(NEW%)<>0 then goto *LIST4
          goto *ENDING
*LIST4:   '或る場所にある駒は斜めに動けません
          CHECK%=NANAME_CHECK(OLD%):if CHECK%=0 then CLIST%=100:goto *ENDING
          CLIST%=4:NEW%=OLD%+15:if OLD%= 87 and NEW%=102 then goto *LIST5
          if BOARD(NEW%)<>0 then goto *LIST5
          goto *ENDING
*LIST5:   CLIST%=5:NEW%=OLD%-15:if OLD%= 102 and NEW%=87 then goto *LIST6
          if BOARD(NEW%)<>0 then goto *LIST6
          goto *ENDING
*LIST6:   CLIST%=6:NEW%=OLD%+17:if OLD%= 38 and NEW%= 55 then goto *LIST7
          if BOARD(NEW%)<>0 then goto *LIST7
          goto *ENDING
*LIST7:   CLIST%=7:NEW%=OLD%-17:if OLD%=55 and NEW%=38 then CLIST%=100:goto *ENDING
          if BOARD(NEW%)<>0 then CLIST%=100:goto *ENDING
          goto *ENDING
*ENDING:COM_MOVE_CHECK=CLIST%
end function
function MAN_MOVE_CHECK(OLD%,NEW%)
var CHECK%
          MAN_MOVE_CHECK=0
          '空白では無い所には移動できません
          if BOARD(NEW%)<>0 then goto *ENDING
          '雪隠の或る場所には移動できません
          if OLD%=102 and NEW%= 87 then goto *ENDING
          if OLD%= 87 and NEW%=102 then goto *ENDING
          if OLD%= 38 and NEW%= 55 then goto *ENDING
          if OLD%= 55 and NEW%= 38 then goto *ENDING
          if OLD%= 86 and NEW%= 87 then goto *ENDING
          if OLD%= 87 and NEW%= 86 then goto *ENDING
          if OLD%= 54 and NEW%= 55 then goto *ENDING
          if OLD%= 55 and NEW%= 54 then goto *ENDING
          if NEW%=OLD%+16 then MAN_MOVE_CHECK=1:goto *ENDING
          if NEW%=OLD%-16 then MAN_MOVE_CHECK=1:goto *ENDING
          if NEW%=OLD%- 1 then MAN_MOVE_CHECK=1:goto *ENDING
          if NEW%=OLD%+ 1 then MAN_MOVE_CHECK=1:goto *ENDING
          '或る場所にある駒は斜めに動けません
          CHECK%=naname_check(OLD%)
          if CHECK%=0 then goto *ENDING
          if NEW%=OLD%+15 then MAN_MOVE_CHECK=1
          if NEW%=OLD%-15 then MAN_MOVE_CHECK=1
          if NEW%=OLD%+17 then MAN_MOVE_CHECK=1
          if NEW%=OLD%-17 then MAN_MOVE_CHECK=1
          *ENDING
end function
function NANAME_CHECK(KOMA%)
          naname_check=1
          '或る場所にある駒は斜めに動けません
          if KOMA%= 35 or KOMA%= 37 or KOMA%= 50 or KOMA%= 52 then NANAME_CHECK=0
          if KOMA%= 54 or KOMA%= 56 or KOMA%= 67 or KOMA%= 69 then NANAME_CHECK=0
          if KOMA%= 71 or KOMA%= 82 or KOMA%= 84 or KOMA%= 86 then NANAME_CHECK=0
          if KOMA%= 88 or KOMA%= 99 or KOMA%=101                then NANAME_CHECK=0
end function
function mousewait()
          while mouse(2,1)<>-1 and mouse(2,0)<>-1
                    if mouse(2,1)=-1 then mousewait=1
                    if mouse(2,0)=-1 then mousewait=0
          wend
          while mouse(2,1)=-1 or mouse(2,0)=-1
          wend
end function
```

★「マイクロプロセッサ誕生30周年企画」良かったです。当時を懐かしく読ませていただきました。またこのような企画をお願いします。（大阪府茨木市・**匿名希望40歳**）……**【編：**この企画の反響が大きいですね。**つぐ美：**若い読者もぜひ読んで，勉強してください。**】**

Windows（HSP V2.5）用



手軽に遊べるカードゲーム

神経衰弱（しんけいすいじゃく） あわせてポン

三好　利治（46歳）

はじめに

チェッカーフラッグで「画像ファイルなしで遊べると良かったのですが」と書かれていましたので，テキストで代用して作り直してみました。

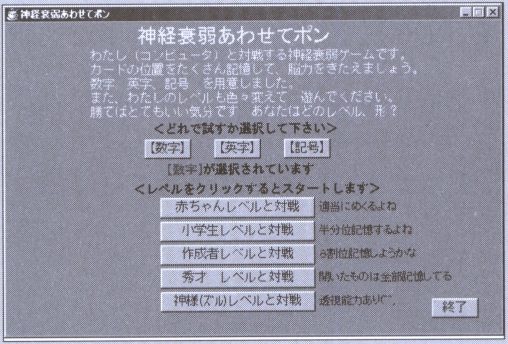
でも，「もう一歩でしたでしょう」ではないので，載らないでしょうけど。

業務ソフトやモジュールならともかく，ゲーム，とくにカードゲームのような静的ゲームのソフト，思考ルーチンなどを作るには，HSPのような言語は，便利で楽だと思います（とくにインタプリタならではの，デバックのしやすさや自由度を考えると）。

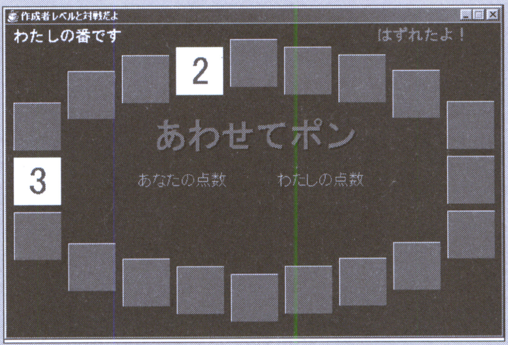
内容

わたし（コンピュータ）と対戦する神経衰弱ゲームです。

これはローカル・ルールがないし，みんな一度は遊んだことがあるはずですので，説明書など読まずに，すぐ見て遊べ

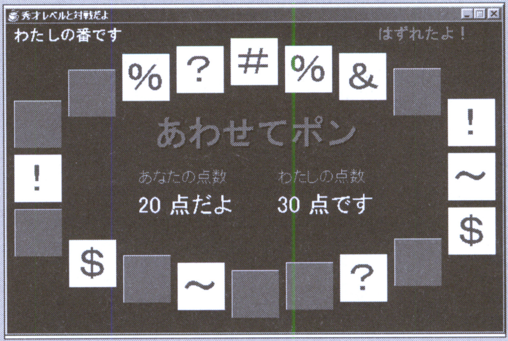


《写真1》複数の絵柄とレベルが選べるのが特徴



《写真2》説明文なしでも，遊べる内容

《写真3》作成者レベルが一番バランスがいい？



《写真4》記号の絵柄は，こんな感じになる

★★★★←黒い三連星（笑）。……といっても，ベーマガの長い歴史の中，このネタ何回送られたんだろう……。（東京都練馬区・ルストフレム14歳）……【影：残念でした。OFの頭の記号も★なので，四連星になってしまいました。そこを見越してないのは甘いぞ。】

ます。

カードの絵柄（今回は絵とは言えませんけど）と位置の両方を記憶する能力が必要になります。子供の能力向上や，私のような歳の人の記憶力確認に役立つ（？）かもしれません。

今回，画像ファイルなしにした代わりというわけじゃありませんが，対戦レベルで「秀才」レベルを増やしました。

で，「赤ちゃん」「小学生」「作者^^;」「秀才」「神様（ズルします）」の五つのレベルでの対戦となります。

勝てばとてもいい気分です。あなたはどのレベル，どの絵（？）が得意なのでしょう。

使用方法

マウスのみ使用します。画面の指示にしたがって，画像をまずクリックして，つぎにレベルを選ぶとゲーム・スタートです。

神経衰弱と同じルールです（ローカル・ルールがないので助かります）。裏返しのカードをマウスでクリックします（2回ね）。コンピュータは，あなたがまちがったら，つぎに勝手にめくりますのでその間は待ってやってくださいね。

神様レベルで勝った場合のみ，画面コピーするか聞いてきますので，YESをクリックすると，自動でデスクトップが画像として残せます。でも，それまでやろうとする人も，勝てる人もいないでしょう。

プログラムの仕組みなど

特別なことは何もしておりませんので，そのまま打ち込むだけです。

思考ルーチンの考えかたは，以下のとおりです。

「赤ちゃん」：なにも考えずに適当にめくります。

「小学生」：すでにめくられたカードを覚えているので，1枚目のカードを覚えていれば，そのカードと同じカードをめくります。

「作者（^^;）：記憶しているカードに2枚の同じ絵のカードがあれば，そのカードをめくります。なければ適当にめくります。2枚目は，記憶している1枚目と同じカードを探しにいく確率を3/4にしました。記憶力75%です。それをめくります。なければ他をめくります。

「秀　才」：記憶しているカードに，2枚の同じ絵のカードがあればそのカードをめくります。なければ適当にめくります。2枚目は記憶している1枚目と同じカードがあれば，100%それをめくります。なければ他をめくります。

「神　様」：めくる前から，カードの裏の絵を見られますので，覗き見してあっているカードをめくります。これじゃあ，身もふたもないですから，たまにまちがえます。

こんな考えでプログラムを書きました。

最後に

改造などは，自分で画像を用意すると（60×60位で名前を統一すると良いですね。扱える画像はbmp，magまたはjpgです），きれいで楽しいものができますし，表示位置（x1,x2,y1,y2の配列）を変えたり，枚数を増やしたり（配列数を増やす），自分のオリジナルができると思います。

HSP

CHECKER FLAG

編：以前に指摘した箇所をしっかりと修正しての採用です。「載らないでしょう」と書いてある作品でも，掲載してしまうのがベーマガです。

Dr.D：神経衰弱なので，ゲーム・システムの面ではまったく新鮮味がないに等しいが，COMの思考ルーチンが数パターンあるのは，なかなかじゃな。神経衰弱のルールであれば，COMの思考ルーチンを作成するのはそんなにむずかしくはないが，それぞれのレベルにしっかりと個性を持たせている部分はよくできている。神様レベルは，かなり卑怯ではあるが，プログラムではこういった方法もあることは知っておいてほしいものじゃ。

つぐ美：その神様レベルに勝ったときに呼び出している“sendtomspaint.exe”は，ダウンロード・ファイルに収録されています。まあ，ほとんどゲームに影響はないですし，必要な［PrtSc］キーで自分でキャプチャできますからね。

「神経衰弱 あわせてポン」のプログラム・リスト

```
;------------------------------------------
; 神経衰弱あわせてポン  awasete2.as v1.00 2001.10 Toshi.miyoshi
;------------------------------------------
        screen 0,640,400,0,dispx-640/2,dispy-400/2
;
        dim x1,20:dim x2,20:dim y1,20:dim y2,20  ;カード表示位置
       x1= 80,150,220,290,360,430,500,570,570,570,500,430,360,290,220,150, 80,10,10,10
       x2=140,210,280,350,420,490,560,630,630,630,560,490,420,350,280,210,140,70,70,70
;
       y1= 60, 40,30,20,30, 40, 60,100,170,240,280,300,310,320,310,300,280,240,170,100
       y2=120,100,90,80,90,100,120,160,230,300,340,360,370,380,370,360,340,300,230,160
;
        sdim suuji,2,11:sdim eiji,2,11:sdim kigou,2,11  ;画像のかわり
        suuji=" "," 0"," 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8"," 9"
       eiji =" "," a"," b"," c"," d"," e"," f"," g"," h"," i"," j"
       kigou=" ","#","$","%","&","*","?","¥","@","!","~"
;
        sdim moji,2,11      ;選択文字格納用
;
        dim card,20         ;カード内容ID
        dim hcard,20        ;開いたカードの記憶用
        dim hiti,20         ;   ""     位置記憶
        dim flg,20          ;当たり済みフラグ
;
;効果音  無くてもいいですけど、あったほうが雰囲気が出ます
        sndload "kan.wav",1:sndload "maru.wav",2
        sndload "batu.wav",3:sndload "yatt.wav",4
        sndload "kekka.wav",5
        mondai=1
;
;---- 説明及び選択画面 -----
*sentaku
        gsel 0:cls
        title "神経衰弱あわせてポン"
        color 115,140,222:boxf
        color 240,240,240:font "ＭＳゴシック",26
        pos 170,5:mes "神経衰弱あわせてポン"
        font "ＭＳゴシック",18
        pos 110,35
        mes "わたし（コンピュータ）と対戦する神経衰弱ゲームです。"
        mes "カードの位置をたくさん記憶して、脳力をきたえましょう。"
```

★報告！初級シスアドに合格しました！（新潟市・whitefox 18歳）……【一同：おめでとうございます。影：ここで，ベーマガ用語解説。「一同」とは編集部全員を指しているのであって，一同さんがいるわけではない。たまに質問が来るので，ここで回答しておいたぞ。】

```
        mes "数字、英字、記号　を用意しました。"
          mes "また、わたしのレベルも色々変えて　遊んでください。"
          mes "勝てばとてもいい気分です　あなたはどのレベル、形？"
          objsize 60,25
          pos 550,355:button "終了",*tmend
;
          color 0,0,0:font "ＭＳゴシック",16,1
          pos 190,128:mes "＜どれで試すか選択して下さい＞"
          pos 180,150:button "【数字】",*tsuji
          pos 270,150:button "【英字】",*tiro
          pos 360,150:button "【記号】",*tzukei
          color 128,0,0
          if mondai=1:pos 200,180:mes "【数字】"
          if mondai=2:pos 200,180:mes "【英字】"
          if mondai=3:pos 200,180:mes "【記号】"
          color 0,0,0:pos 255,180:mes "が選択されています"
;
          pos 165,205:mes "＜レベルをクリックするとスタートします＞"
          objsize 200,26
          pos 200,225:button "赤ちゃんレベルと対戦",*tlv1
          pos 200,255:button "小学生レベルと対戦",*tlv2
          pos 200,285:button "作成者レベルと対戦",*tlv3
          pos 200,315:button "秀才　レベルと対戦",*tlv4
          pos 200,345:button "神様(ｽﾞﾙ)レベルと対戦",*tlv5
;
          font "ＭＳゴシック",14
          pos 405,230:mes "適当にめくるよね"
          pos 405,260:mes "半分位記憶するよね"
          pos 405,290:mes "８割位記憶しようかな"
          pos 405,320:mes "開いたものは全部記憶してる"
          pos 405,350:mes "透視能力あり(^^;"
          stop
*tsuji
          color 115,140,222:boxf 200,180,255,200
          mondai=1:color 128,0,0:font "ＭＳゴシック",16,1
          pos 200,180:mes "【数字】"
          stop
*tiro
          color 115,140,222:boxf 200,180,255,200
          mondai=2:color 128,0,0:font "ＭＳゴシック",16,1
          pos 200,180:mes "【英字】"
          stop
*tzukei
          color 115,140,222:boxf 200,180,255,200
          mondai=3:color 128,0,0:font "ＭＳゴシック",16,1
          pos 200,180:mes "【記号】"
          stop
*tlv1
          lv=1:tit="赤ちゃんレベルと対戦だよ":goto *start
*tlv2
          lv=2:tit="小学生レベルと対戦だよ":goto *start
*tlv3
          lv=3:tit="作成者レベルと対戦だよ":goto *start
*tlv4
          lv=4:tit="秀才レベルと対戦だよ":goto *start
*tlv5
          lv=5:tit="神様(ズル)レベルと対戦だよ":goto *start
*tmend
          end
;
;ゲームスタート
*start
          cls
          repeat 11
            if mondai=1:moji.cnt=suuji.cnt
            if mondai=2:moji.cnt=eiji.cnt
            if mondai=3:moji.cnt=kigou.cnt
          loop
```

```
          randomize
;
          color 0,0,128:boxf
          title tit
          font "ＭＳゴシック",44,1
          color 0,0,0    :pos 192,122:mes "あわせてポン"
          color 128,128,0:pos 190,120:mes "あわせてポン"
;
          repeat 20
            flg.cnt=0:hcard.cnt=99:hiti.cnt=99
          loop
          nokori=20:ten1=0:ten2=0
;
          repeat 10
            j=cnt*2:i=j+1
            card.j=cnt+1:card.i=cnt+1
          loop
;
          gsel 0
;
          repeat 20
            color 255,255,255:boxf x1.cnt,y1.cnt,x1.cnt+60,y1.cnt+60
            font "ＭＳゴシック",50:color 0,0,255
            pos x1.cnt+4,y1.cnt+4
            j=card.cnt
            mes moji.j
          loop
;
          color 0,0,128:boxf 170,190,400,290
          color 255,0,255:font "ＭＳゴシック",48
          pos 200,190:mes "まぜるよー"
          repeat 20            :カードをまぜる処理
            rnd j,20
            hozonn=card.cnt
            card.cnt=card.j
            card.j=hozonn
          loop
          wait 100
;
          repeat 20            :カードをふせる
            color 0,128,0:boxf x1.cnt+1,y1.cnt+1,x1.cnt+60,y1.cnt+60
            wait 1
          loop
          color 0,0,128:boxf 170,190,480,290
          color 192,192,192:font "ＭＳゴシック",20
          pos 170,190:mes "あなたの点数"
          pos 350,190:mes "わたしの点数"
;
;
*mekuru ;あなた入力
          wait 10
          color 0,0,128:boxf 480,5,640,30
          color 0,0,128:boxf 10,5,170,30
          color 255,255,255:font "ＭＳゴシック",20,1
          pos 10,5:mes "あなたの番です"
          try=1:han1=99:han2=99
          gosub *anata
          try=2
          gosub *anata
          if han1=han2{
            wait 25
            color 0,0,128:boxf 480,5,640,30
            color 255,0,255:pos 480,5:mes "当たりだよ！"
            snd 2
            nokori=nokori-2:ten1=ten1+1:ten=ten1*10
            gosub *kiokucl
            gosub *tennsuu
            if nokori=0:gosub *kekka:goto *sentaku: else :goto *mekuru
```


★新成人へ一言。大人になるというのは，歳をとること。ソコ！寝ない！（愛知県名古屋市・**最馬場可？歳**）……【**影**：はっ！失礼しました。プログラム・チェック，チェック！うん？誰だ，心地よい昼寝を邪魔したのは。まったく！**編集長**：まったく！とは何だ？！】

```
          }
          else{
            wait 25
            color 0,0,128:boxf 480,5,640,30
            color 255,0,255:pos 480,5:mes "はずれたよ！"
            snd 3:wait 25
            gosub *kiokusuru
            gosub *uragaesu
            flg.clflg1=0
            flg.clflg2=0
            goto *mekuru2
          }
;
*mekuru2  ;ｺﾝﾋﾟｭｰﾀ入力
          wait 40
          color 0,0,128:boxf 480,5,640,30
          color 0,0,128:boxf 10,5,170,30
          color 255,255,255:font "ＭＳゴシック",20,1
          pos 10,5:mes "わたしの番です"
          han1=99:han2=99
          gosub *watasi1
          gosub *watasi2
          if han1=han2{
            wait 25
            color 0,0,128:boxf 480,5,640,30
            color 255,0,255:pos 480,5:mes "当たりだよ！"
            snd 2
            nokori=nokori-2:ten2=ten2+1:ten=ten2*10
            gosub *kiokucl
            gosub *tennsuu2
            if nokori=0:gosub *kekka:goto *sentaku: else :goto *mekuru2
          }
          else{
            wait 25
            color 0,0,128:boxf 480,5,640,30
            color 255,0,255:pos 480,5:mes "はずれたよ！"
            snd 3:wait 25
            gosub *kiokusuru
            gosub *uragaesu
            flg.clflg1=0
            flg.clflg2=0
            goto *mekuru
          }
;
          stop
;
;―――――――――――――――サブルーチン――――――――――――――――
*anata  ;あなた入力
*rkey
          buf=0
          await :stick key,0
          if key!256 :goto *rkey   ;マウスボタン256
          repeat 20                 ;どの位置かチェック
            if (mousex>=x1.cnt)&(mousex<x2.cnt)&(mousey>=y1.cnt)&(mousey<y2.cnt){
              if flg.cnt=1{
                buf=0
                break
              }
              else{
              buf=card.cnt
                xx=x1.cnt:yy=y1.cnt
                flg.cnt=1
                t=cnt
                break
            }
            }
          loop
          if buf=0:goto *rkey
```

```
;
          snd 1                              ;カードを開ける
          color 255,255,255                  ;ように見せる
          repeat 10
            y=6*cnt+6
            boxf xx,yy,xx+60,yy+y
            await 10
          loop
          font "ＭＳゴシック",50
          color 0,0,255:pos xx+4,yy+4:mes moji.buf
;
          if try=1:han1=buf:urax1=xx:uray1=yy:clflg1=t
          if try=2:han2=buf:urax2=xx:uray2=yy:clflg2=t
;
          font "ＭＳゴシック",20,1
          return
;
*watasi1  ;わたし入力１
          color 0,0,128:boxf 480,5,640,30
          color 255,0,255:pos 480,5:mes "どれにするかな"
          wait 100
          buf=0
          if (lv!3)&(lv!4):goto *rtry1
;
;作成者の思考（記憶したカードに２枚同じのがあるば一応それをめくる）
;秀才君の思考（記憶したカードに２枚同じのがあれば100%それをめくる）
          cnt1=-1
*jloop
          cnt2=-1
          cnt1=cnt1+1
          if cnt1>19:goto *jskip
          if hcard.cnt1=99:goto *jloop
*jloop2
          await
          cnt2=cnt2+1
          if cnt2>19:goto *jloop
          if cnt1=cnt2:goto *jloop2
        if hcard.cnt1=hcard.cnt2 {
            t=hiti.cnt2
            buf=hcard.cnt2:xx=x1.t:yy=y1.t:flg.t=1
            goto *jskip
          }
          goto *jloop2
*jskip
          if buf!0:goto *skip1  ;あったから表示し２回目は当てる
;
;赤ちゃんと小学生と神様用（作成者、秀才は該当無い時下記と同じ思考）
*rtry1
          rnd t,20
          if flg.t=1:goto *rtry1
;
        buf=card.t
          xx=x1.t:yy=y1.t
          flg.t=1
;
*skip1
          snd 1                              ;カードを開ける
          color 255,255,255                  ;ように見せる
          repeat 10
            y=6*cnt+6
            boxf xx,yy,xx+60,yy+y
            await 10
          loop
          font "ＭＳゴシック",50
          color 0,0,255:pos xx+4,yy+4:mes moji.buf
;
          han1=buf:urax1=xx:uray1=yy:clflg1=t
;
```


★我校のパソコンにはネットの番犬が住んでいる。はっきり言ってジャマだ。（東京都練馬区・**ロングリッド15歳**）……【**編**：ちなみに編集部の番犬は，いつも寝てばかりいて，役に立ちません。**影**：それって，私のこと？**編**：いやいや，そんな。**影**：あっそう。】

```
          font "MSゴシック",20,1
          return
;
*watasi2  ;わたし入力２
          color 0,0,128:boxf 480,5,640,30
          color 255,0,255:pos 480,5:mes "つぎはねぇ"
          wait 150
          if lv=1:goto *rtry2       ;赤ちゃんは無条件でランダムにめくるへ
          buf=0
;
;神様の思考(^^;
          if lv=5{                          ;神様は覗き１枚目と同じのを探す
            rnd ff,5:if ff=0:goto *rtry2   ;たまにはずす (1/5の確率)
            repeat 20
              if card.cnt=han1{
                 if flg.cnt!1:  buf=han1:xx=x1.cnt:yy=y1.cnt:flg.cnt=1:break
              }
            loop
            goto *skip2  ;無理やり当てて表示へ
          }
;
;作成者の思考（２枚目はうろ覚え、結果として７５％の記憶力）
          if lv=3 {
             rnd ff,4
             if ff=0:goto *rtry2
          }
;
;小学生の思考、秀才の思考（１枚目と同じカードを探す）
          repeat 20
            if hcard.cnt=han1{
               t=hiti.cnt
               if flg.t!1: buf=han1:xx=x1.t:yy=y1.t:flg.t=1
               if buf!0:break
            }
          loop
          if buf!0:goto *skip2    ;あったから当てる
;
*rtry2                            ;該当無いのでランダムにめくる
          buf=0
          rnd t,20
          if flg.t=1:goto *rtry2
;
        buf=card.t
          xx=x1.t:yy=y1.t
          flg.t=1
;
*skip2
          snd 1                             ;カードを開ける
          color 255,255,255                 ;ように見せる
          repeat 10
            y=6*cnt+6
            boxf xx,yy,xx+60,yy+y
            await 10
          loop
          font "MSゴシック",50
          color 0,0,255:pos xx+4,yy+4:mes moji.buf
;
          han2=buf:urax2=xx:uray2=yy:clflg2=t
;
          font "MSゴシック",20,1
          return
;
*kiokusuru  ;開けたカードを記憶する
;
          sw1=0:sw2=0
          repeat 20
            if (hcard.cnt=han1)&(hiti.cnt=clflg1):sw1=1
            if (hcard.cnt=han2)&(hiti.cnt=clflg2):sw2=1
          loop
          if sw1=1:goto *skip4
          repeat 20
            if hcard.cnt=99:hcard.cnt=han1:hiti.cnt=clflg1:break
          loop
*skip4
          if sw2=1:goto *skip5
          repeat 20
            if hcard.cnt=99:hcard.cnt=han2:hiti.cnt=clflg2:break
          loop
*skip5
          return
;
*kiokucl  ;当たったカードの記憶を消す
;
          repeat 20
          if hcard.cnt=han1:hcard.cnt=99:hiti.cnt=99:break
          loop
          repeat 20
            if hcard.cnt=han2:hcard.cnt=99:hiti.cnt=99:break
          loop
          return
;
*uragaesu  ;開けて外れたカードを裏返す
;
          color 0,128,0
          repeat 30
            y=cnt*2+2
            boxf urax1+1,uray1+1,urax1+60,uray1+y
            await 20
          loop
;
          repeat 30
            y=cnt*2+2
            boxf urax2+1,uray2+1,urax2+60,uray2+y
            await 20
          loop
          return
;
*tennsuu  ;あなた点数表示
          color 0,0,128:boxf 160,220,290,290
          color 255,255,255:font "MSゴシック",30
          pos 160,220:mes " "+ten+" 点だよ"
          return
;
*tennsuu2  ;わたし点数表示
          color 0,0,128:boxf 340,220,470,290
          color 255,255,255:font "MSゴシック",30
          pos 340,220:mes " "+ten+" 点です"
          return
;
*kekka  ;結果発表
          snd 5
          if (lv=5)&ten1>ten2:gosub *gamencopy:return ;神様レベルに勝った場合のみ,
                                                    ;まずければコメントにして下さ
い
;
          if ten1=10:snd 4:dialog "１００点！  完勝です  おめでとう(^o^",0:return
          if ten1>ten2:snd 4:dialog "あなたの勝ちです  おめでとう！(^o^",0:return
          if ten1=ten2:dialog "同点でしたね",0:return
          dialog "あなたの負けです！  ぽんぽこぽん(^^;",1:return
;
*gamencopy ;おまけの余計なお世話処理
          snd 4
          dialog "神様に勝ちました！¥nおめでとうございます¥n記念に画面キャプチャーします？",2
          if stat=6:exec "sendtomspaint.exe",1
          return
;
;====================================//END OF DATA
```

★問題集を開きつつ，その上でOFを書いている自分がいる…。（東京都練馬区・**どらねく15歳**）……【**影**：こらこら，勉強するときは勉強，遊ぶときは遊ぶ，OF書くときはボーッとする。しっかりメリハリをつけましょう。**編集長**：よし，自分に言い聞かせてるんだな。】

Windows（HSP V2.5）用

あばよ，とっつぁ～ん

ルパン

くろすけ（34歳）

ゲームの説明

あなたは天下の大泥棒ルパ～ン三世。誰にも捕まらずにすべての金貨を盗んでください。

$袋を取ると一定時間，影になって敵を素通りできます。また，$袋を残してステージ・クリアすると，残した数が蓄積されてゆき，10個たまるごとにルパンの残り数が増えます。

ルパンの敵

・**ゼニガタ警部**…ニッポンの警察官。地の果てまでもルパンを追い続ける男。

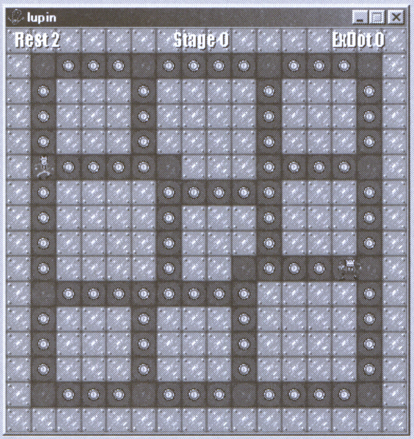
・**ガードマン**…ビルのガードマン。日本じゃないので銃も持ってるぞ。

・**フジ子ちゃん**…ある時は盗賊，ある時は女スパイ。ルパンを骨抜きにするセクシー美女。

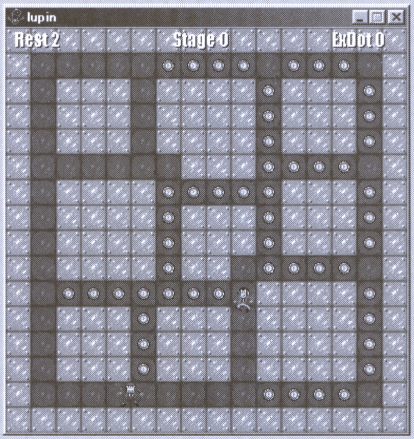
・**伯爵**…家宝をルパンに奪われ，恨んでいる。「ジョドオォォッ！」

ルパンの操作方法

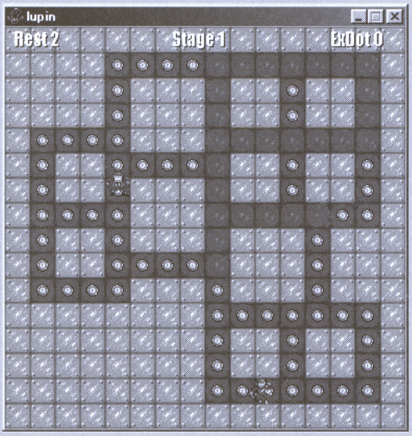
カーソル・キーで上下左右に移動します。



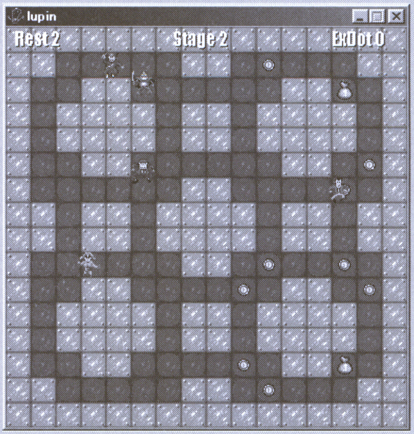
《写真1》某有名マンガとそっくりの設定



《写真2》画面にある金貨をすべて盗め



《写真3》影になれば敵キャラをすり抜けられる



《写真4》多彩なステージも、このゲームの特徴

★影さんに質問！パソコンで雪合戦できるのか？（愛知県江南市・**甲斐俊夫29歳**）……【**影**：人間がパソコンを投げ合う場合は「パソコン合戦」。パソコンに雪を投げさせるのは至難の技だが，パソコンの着ぐるみを着た人間が投げ合うことは可能。その場合「雪合戦」。】

《リスト1》「ルパン」のプログラム・リスト

```
        title "lupin"

        dot=24 :xys=dot*16                                  ; ﾒｲﾝ画面(ID0)
        screen 0,xys,xys,1,dispx-xys/2,dispy-xys/2          ;
        font "Impact",18                                    ;

        cpt=dot*2                                 ; ﾊﾞｯﾌｧ画面(ID2)
        buffer 2,xys,cpt+xys,1                              ;
        font "Impact",18                          ;
        picload "chrs.bmp",1,0,0                  ;

        sndload "eat.wav",0                       ; 効果音
        sndload "exp.wav",1                       ;
        sndload "los.wav",2                       ;

        sdim txbf,512 :sdim lbf,512
        alloc mm,256
*start
        stg=0 :my=3 :exp=0
*stgr                                                       ; ｽﾃｰｼﾞﾃﾞｰﾀ読込み
        fn="stage"+stg+".txt" :exist fn
        if strsize<0 { x=xys/2-40 :y=xys/2-36 :s="Complete" :gosub *mojp
                x=xys/2-60 :y=xys/2+18 :s="Congratulation!" :gosub *mojp
                goto *repwt
        }
        bload fn,txbf :notesel txbf
        noteget lbf,16
        getstr nrm,lbf,5,',' :int nrm             ; ｸﾘｱｰﾉﾙﾏ数
        getstr en,lbf,5+strsize :int en                     ; 敵数

        noteget lbf,17 :i=5
        getstr px,lbf,i,',' :i+=strsize :int px :px=px*4    ; ﾌﾟﾚｲﾔｰ座標
        getstr py,lbf,i,',' :i+=strsize :int py :py=py*4    ;
        getstr pdr,lbf,i :int pdr                                     ;

        repeat en
                noteget lbf,18+cnt :i=5
                getstr a,lbf,i,',' :i+=strsize :int a :ex.cnt=a*4  ; 敵座標
                getstr a,lbf,i,',' :i+=strsize :int a :ey.cnt=a*4  ;
                getstr a,lbf,i :int a :edr.cnt=a                             ;
        loop

        gsel 2 :gmode 1,dot,dot                             ; ﾊﾞｯﾌｧ画面にｽﾃｰｼﾞ描画
        repeat 16 :y=cnt
                repeat 16 :x=cnt
                        peek a,txbf,18*y+x :a-=48
                        pos dot*x,dot*y+cpt :gcopy 2,dot*a,0
                        poke mm,16*y+x,a                              ; mmをﾊﾞｲﾅﾘﾏｯﾌﾟﾃﾞｰﾀにする
                loop
        loop
        x=8 :y=cpt :a=my-1 :s="Rest "+a :gosub *mojp
        x=xys/2-32 :s="Stage "+stg :gosub *mojp
        x=xys-72 :s="ExDot "+exp :gosub *mojp

        gsel 0,1 :cls 4 :palcopy 2                ; ﾊﾞｯﾌｧ画面のｽﾃｰｼﾞ画像をﾒｲﾝ画面にｺﾋﾟｰ
        pos 0,0 :gcopy 2,0,cpt,xys,xys            ;

        gmode 2,dot,dot :dv=dot/4
        pos px*dv,py*dv :gcopy 2,dot*4,0                              ; ﾌﾟﾚｲﾔｰと敵を配置
        repeat en :c=cnt                                              ;
                pos ex.c*dv,ey.c*dv :gcopy 2,dot*2*c,dot ;
        loop

        wait 300 :tm=0
*main
        redraw 0

        a=0                                                 ; 敵につかまった?
```

プログラムの入力方法

まず，絵と音のデータを作成しましょう。

・**キャラクター画像**：chrs.bmp，192×48ドット（1キャラ=24×24ドット）

・**効果音データ**は，eat.wav：金貨ゲット，exp.wav：＄袋ゲット，los.wav：捕まった！，となります。

つぎにステージ・データを入力し，作成します。

最後にプログラムを入力し，[コンパイル＋実行]すれば，とりあえず遊べます。

EXEファイルの作りかたについては，HSPのマニュアルを見てね。

ステージの追加について

ステージ・データはテキスト形式で，いくつでも追加できます。

ステージは16×16ブロックで，数値＝キャラクターの対応は以下のとおりです。

0=壁，1=＄袋，2=金貨，3=床

パラメータは，行を空けずにステージ・データに続けて記述してください（**第1表**）。

追加するファイル名は，stage8.txt stage9.txt stage10.txt … と付けていってください。

最後に一言

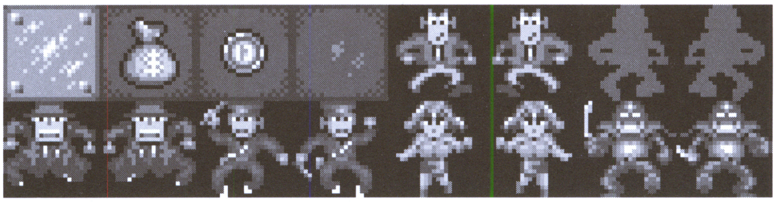
影になって素通りできるのは，ルパンならではの技…ということで。

CHECKER FLAG

編：権利面での扱いが，かな～り微妙な作品ですが，ゲーム自体の完成度を考慮しての採用となりました。

Dr.D：基本はドットイット・ゲームというスタンダードな形なので新鮮味はないが，敵キャラやステージ構成などがしっかり作られておる。影になって素通りできるというのは，おもしろいアイデアじゃな。

影：ん，Dr.私のこと呼びました？



《第1図》chrs.bmpの画像

《第1表》ステージのパラメータの説明

NrEn	クリアとなる金貨数,敵の数
MyPo	プレイヤーX座標(0～15),プレイヤーY座標(0～15),プレイヤー移動方向(0～3)
EnPo	敵1のX座標(0～15),敵1のY座標(0～15),敵1の移動方向(0～3)
EnPo	敵2のX座標(0～15),敵2のY座標(0～15),敵2の移動方向(0～3)
EnPo	敵3のX座標(0～15),敵3のY座標(0～15),敵3の移動方向(0～3)
EnPo	敵4のX座標(0～15),敵4のY座標(0～15),敵4の移動方向(0～3)

★次号発売日を目次に書いてください。（岡山市・**楽道31歳**）……【**影**：ベーマガは毎月8日発売ということはいつも頭に入れておいてくださいね。休みの関係で早まる場合はありますけど。いずれにしても，目次に次号発売日は入れるようにしましょう。】

```
        repeat en :c=cnt
                if (px+3>ex.c)&(px-3<ex.c)&(py+3>ey.c)&(py-3<ey.c)&(tm=0) :a=1
        loop
        if a=1 :snd 2 :goto *nxtstg

        if (px¥4=0)&(py¥4=0) { x=px/4 :y=py/4 :g=16*y+x
                peek a,mm,g
                if (a=1)|(a=2) :gosub *doteat               ; 金貨ゲット

                stick ky,15                                          ; プレイヤー移動方向
                peek a,mm,g-1 :if (ky=1)&(a!0) :pdr=3                ;
                peek a,mm,g+1 :if (ky=4)&(a!0) :pdr=1                ;
                peek a,mm,g-16 :if (ky=2)&(a!0) :pdr=0               ;
                peek a,mm,g+16 :if (ky=8)&(a!0) :pdr=2               ;

                if pdr¥2=0 :b=16*(pdr-1)+g :else :b=2-pdr+g
                peek a,mm,b :if a=0 :goto *skp
        }
        if pdr¥2=0 :py=py+pdr-1 :else :px=px+2-pdr          ; プレイヤー移動処理
*skp
        gmode 1,xys,xys :pos 0,0 :gcopy 2,0,cpt              ; 背景(ステージ)描画

        gmode 2,dot,dot
        a=px+py/4¥2 :if tm>0 :tm- :a+=2                      ; a=プレイヤーキャラパターン
        pos px*dv,py*dv :gcopy 2,dot*a+(dot*4),0 ; プレイヤー描画

        repeat en :c=cnt
                if (ex.c¥4=0)&(ey.c¥4=0) { x=ex.c/4 :y=ey.c/4 :g=16*y+x
                        rnd r,2 :r=2*r+1 :r=edr.c+r¥4                ; ランダム移動方向
                        if r¥2=0 :b=16*(r-1)+g :else :b=2-r+g
                        peek a,mm,b :if a!0 :edr.c=r :goto *skp2
                        if edr.c¥2=0 :b=16*(edr.c-1)+g :else :b=2-edr.c+g; 移動方向が壁なら
                        peek a,mm,b :if a=0 :edr.c=edr.c+2¥4         ; 方向転身させる
                }
*skp2
                if edr.c¥2=0 :ey.c=ey.c+edr.c-1 :else :ex.c=ex.c+2-edr.c ; 敵移動処理

                a=ex.c+ey.c/4¥2 :b=dot*2*c                   ; a=敵キャラパターン
                pos ex.c*dv,ey.c*dv :gcopy 2,dot*a+b,dot ; 敵キャラ描画
        loop

        redraw 1 :await 60

        if nrm>0 :goto *main
*nxtstg
        redraw 1 :x=xys/2-43 :y=xys/2-9
        if nrm<1 { s="Stage Clear" :gosub *mojp             ; ノルマチェック
                stg+ :c=0
                repeat 256                                   ; $袋の残り数チェック
                        peek a,mm,cnt :if a=1 :c+            ;
                loop                                         ;
                exp+=c :if exp>9 :my+ :exp-=10               ; 10個でプレイヤー+1
        } else :my-

        wait 300
        if my>0 :goto *stgr                          ; 残プレイヤーあればステージへ
        s=" Game Over" :gosub *mojp
*repwt
        stick ky :wait 1 :if ky&48 :goto *start              ; キー入力待ち
        goto *repwt
*doteat
        gsel 2 :pos px*dv,py*dv+cpt :gcopy 2,dot*3,0         ; 金貨を消す描画
        poke mm,g,3
        if a=2 :nrm- :snd 0 :else :tm=120 :snd 1 ; $袋なら無敵タイムセット
        gsel 0 :return
*mojp
        color 0,0,0 :pos x+1,y+1 :mes s                      ; 浮き文字ルーチン
        color 255,255,255 :pos x,y :mes s
        return
```

《リスト2》ステージ0

```
0000000000000000
0322232222322230
0200020000200020
0200020000200020
0200020000200020
0322223000222230
0200002222200020
0200002000200020
0200002000200020
0200002003222230
0322222222000020
0200020002000020
0200020002000020
0200020002000020
0322232223222230
0000000000000000
NrEn=82,1
MyPo=1,5,3
EnPo=14,9,0
```

《リスト3》ステージ1

```
0000000000000000
0000222222222220
0000200020020020
0000200020020020
0222200022212220
0200222220020020
0200200020020020
0222200022222220
0200200020002000
0200222220002000
0222200020002000
0000000022222220
0000000020020020
0000000020020020
0000000022222220
0000000000000000
NrEn=90,2
MyPo=8,7,3
EnPo=12,11,0
EnPo=4,5,1
```

《リスト4》ステージ2

```
0000000000000000
0033323003233300
0310032332300130
0300003003000030
0300003003000030
0230032332300320
0323323003233230
0030030000300300
0030030000300300
0323323003233230
0230032332300320
0300003003000030
0300003003000030
0310032332300130
0033323003233300
0000000000000000
NrEn=24,4
MyPo=2,2,1
EnPo=6,6,0
EnPo=9,6,1
EnPo=9,9,2
EnPo=6,9,3
```

《リスト5》ステージ3

```
0000000000000000
0122200000022220
0200200000020020
0200200000020020
0222222222222220
0200002002000020
0200002002000020
0200002002000020
0222222221222220
0200020002000200
0200020002000200
0222220002222200
0200020002000200
0200020002000200
0122222222222200
0000000000000000
NrEn=92,3
MyPo=14,8,1
EnPo=3,1,3
EnPo=6,8,1
EnPo=12,14,3
```

《リスト6》ステージ4

```
3000000000000003
0022222002222200
0210002222000120
0200021001200020
0202220330222020
0222020330202220
0202022002202020
0222002222002220
0200002002000020
0200012002100020
0222220000222220
0202022002202020
0122002222002210
0002200000022000
3300222222220033
3330000000000333
NrEn=100,4
MyPo=6,9,1
EnPo=7,5,0
EnPo=9,9,0
EnPo=8,4,2
EnPo=6,14,1
```

《リスト7》ステージ5

```
0000000000000000
0122222222222210
0200020000200020
0202222222222020
0202002002002020
0222012222102220
0202220000222020
0202020110202020
0202020110202020
0202220000222020
0222012222102220
0202002002002020
0202222222222020
0200020000200020
0122222222222210
0000000000000000
NrEn=116,4
MyPo=3,3,0
EnPo=9,5,0
EnPo=10,6,1
EnPo=9,10,2
EnPo=5,9,3
```

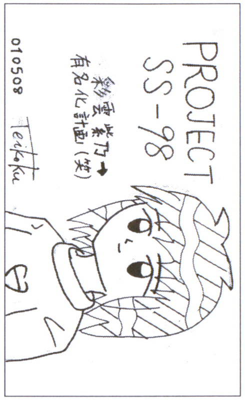
《リスト8》ステージ6

```
0000000000000000
0022222222222200
0220020000200220
0200220310220020
0202200300222020
0222222222202220
0200020020200020
0201022220233020
0203320222201020
0200020200200020
0222022222222220
0202220030022020
0200220130220020
0220020000200220
0022222222222200
0000000000000000
NrEn=112,4
MyPo=5,5,0
EnPo=8,3,0
EnPo=12,8,1
EnPo=7,12,2
EnPo=3,7,3
```

《リスト9》ステージ7

```
0000000000000000
0333333333330300
0302020202033330
0322222222230030
0302020202030330
0322222222230300
0302020202030330
0322222222230030
0302020202030330
0322222222230030
0302020202030330
0333333333330300
0000000000000330
0333333330033300
0130303033330330
0000000000000000
NrEn=56,4
MyPo=13,1,0
EnPo=3,5,0
EnPo=1,14,1
EnPo=7,7,2
EnPo=1,11,3
```

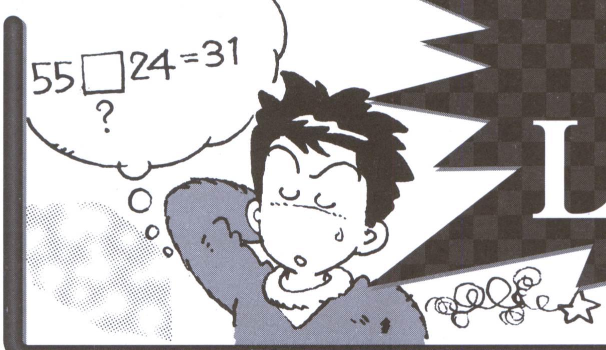
評：横イラスト少ーない（編）



by ていこく

★はがきとネットはがきの質問がちがうけど，いいのかしらん。(千葉県松戸市・**ほおく26歳**)……【**つぐ美**：細かいこと言わないの。**影**：じつは，誰か指摘してくるのを待っていたんだよ。**編**：ということで，次回のWeb更新をお楽しみに。**影**：私が更新するの？】

iアプリ用



iモードで頭の体操？

移植版

L-Quest for i
(ロークエスト フォー アイ)

小倉　邦治（20歳）

内容

本誌2000年6月号掲載の，零戦闘機使用MIDDLEさん作 PC-G850(ポケコン)用『L-Quest』をiアプリへ移植したものです。

画面に表示される計算式で使用されている演算子を，時間内に当てていくゲームです。

遊びかた

左右キーと決定キーで操作します。左右キーで演算子を選択します。現在選択している演算子が赤く表示されます。

決定キーで，選択した演算子に決定し，その結果が画面に表示されます。「Good!」なら正解でScoreが＋10，「Bad!」なら不正解でScoreが－10されます。3回まちがえるとゲーム・オーバーです。

再始，Exitするには，対応しているソフトキーを押します。

また画面には，問題式や選択する演算子の他，Score(得点)，Hit(連続正解数)，Timer(制限時間)，Q(問題数)などが表示されています。Hitは一度まちがえると初期化されます。

入力方法

Main.java(**リスト1**)，LQuest.java(**リスト2**)をテキスト・エディタ(メモ帳でもOK)で入力します。入力し終えたら，各環境に合わせてビルドを行います。

NTT公認のiアプリ開発ツールDoJaを使えば，ボタン一つでビルドができます。また，**リスト3**のjamファイルの制作も楽に行えます。

原作とのちがいなど

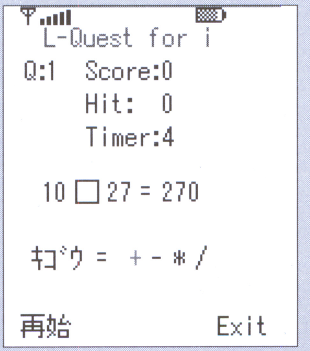
ポケコン版を遊んでいないので，くわしくはわかりませんが，ゲーム構成が異なっているはずです(ポケコン版にあるステージがない，など)。

連続して問題を解いていく形式にしたことで，暇つぶし感覚で遊べる形になったとは思います。

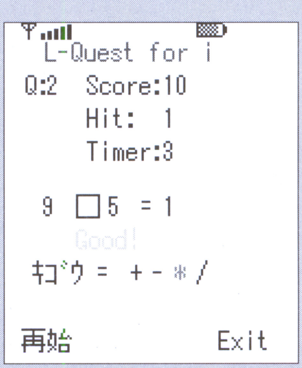
CHECKER FLAG

Dr.D：ポケコン版で指摘した点を，しっかり修正しているのが評価できるな。ゲームそのものは，移植版ということで新鮮味はないが，iアプリ用のアレンジはしているみたいじゃな。リストが単純なので，説明文で述べているような改造も行いやすいじゃろう。

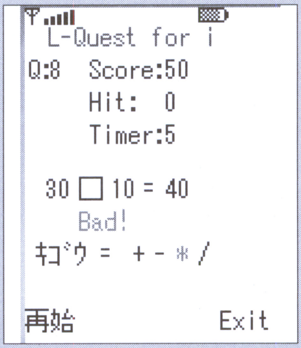
編：チョット手間ですが，問題式の出題方法を工夫するとおもしろそうですね。いかにも引っかけという問題ばかりが出題されるなどの工夫ができそうです。



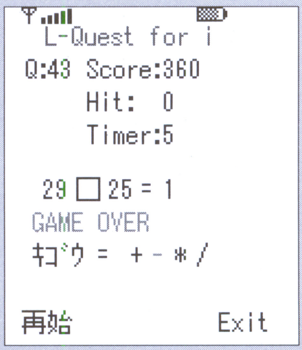
《写真1》計算で使用した演算子を当てる



《写真2》問題自体はそんなにむずかしくないが



《写真3》調子に乗るとけっこうまちがえがち



《写真4》連続正解の最高記録も表示されると良かった？

★ベーマガと○経プログラミングを愛読していますが，おもしろさ，わかりやすさは，ベーマガのほうがワンランク上だと思います。これからも楽しいページをお願いしますね！（島根県松江市・NSR250R十横浜ナンバー37歳）……【影：おっほん！おまかせください！】

《リスト2》「移植版　L-Quest for i」LQuest.javaのプログラム・リスト

```
import com.nttdocomo.ui.*;
import java.util.*;

public class LQuest extends Canvas {

	//変数の宣言
	int[] nowcolor = new int[4];
	int[] ans = new int[4];
	int n1,n2,p,q,cur;
	int setf,ok,hit,endck,key;
	int score;
	ShortTimer timer;
	int qtime;
	//乱数の初期化
	Random rnd = new Random();

	//コンストラクタ(キーの名前設定を行う)
	public LQuest() {
		setSoftLabel(Frame.SOFT_KEY_1, "再始");
		setSoftLabel(Frame.SOFT_KEY_2, "Exit");

	timer = ShortTimer.getShortTimer(this,0,1000,true);

		Init(0);

	}

	//ゲーム関連の初期化
	public void Init(int f){
		int i;

		n1=0; n2=0; p=0;
		setf=1; qtime=5;
		//ゲーム開始時に初期化する変数
		if(f==0){
			key=1;
			endck=0;
			score=0;
			hit=0;
			cur=0;
			q=1;
			ok=-1;
			nowcolor[0]=1;
			for(i=1;i<4;i++)
				nowcolor[i]=0;

		}
	}

	//問題の設定
	public void q_set(){
		int i,j,f;

		do{
			f=0;
			//問題の作成と各演算子での答え
			n1 = Math.abs(rnd.nextInt()%30)+1;
			n2 = Math.abs(rnd.nextInt()%30)+1;
			p = Math.abs(rnd.nextInt()%4);
			ans[0] = n1 + n2;
			ans[1] = n1 - n2;
			ans[2] = n1 * n2;
			ans[3] = n1 / n2;
			//同じ値の答えがないかをチェック
			for(i=0;i<4;i++){
				if(i!=p) {
					if(ans[i]==ans[p])
						f=1;
				}
			}
		}while(f==1);
	}

	//正解かどうかの判定
```

iアプリ

また，このポケコン版のCHECKER FLAGで指摘されていた“80□1=1”問題を，Dr.Dが言っている方法で対処してみました。

あとがき

このゲーム，基本的な部分はすぐにできたのだけど、実行してみるとたま〜に異常終了するという変な現象にぶつかりました。

結果的には，問題式を作る部分の関数(q_set)の問題再定義に立てるフラグ(int変数f)を1にしっぱなしで，無限ループに入っていたのと，問題式の二つ目の項に0が入ると割り算のときに0除算を起こしていたという2点が原因でした。う〜ん単純なミスですね。ですが，しばらくその原因に気が付かなかった。エミュレータが悪いんじゃないかって思ったり。たまにバグるってのが，やっかいでしたね。

また，このゲーム，何回かプレイしていて思ったのが，難易度調整の奥が深いということ。というのも出題する問題式の項の範囲によっても，かなり難易度に差がでる。最初は1〜99までの範囲にしてたけど，それだと掛け算の値だけが，かなり大きくなってすぐにわかってしまう。そこで値の範囲を狭めてみました。これくらいの範囲(1〜29)のほうが案外，足し算と掛け算をまちがえたりします。

あと，ゲーム・オーバー判定も、原作どおり一度のミスで終わりってほうが緊張感があるかなっとは思ったけど，iアプリ用ってことで暇つぶし感覚で遊べるよう3回ミスまでプレイできるようにしてみました。

その他にも，チョットしたこと(たとえばタイムの時間設定や，引き算と一目でわからないように，答えにマイナスが出ないように調整する，など)でけっこう印象が変わるゲームです。興味のある人はお好みの改造をしてみてはいかがでしょうか？

最後に，このゲームの原作者，零戦闘機使用MIDDLEさんにお礼を申し上げます。ありがとうございました。

《リスト1》「移植版　L-Quest for i」Main.javaのプログラム・リスト

```
import com.nttdocomo.util.*;
import com.nttdocomo.ui.*;

public class Main extends IApplication {

	static IApplication app;
	static LQuest mainCanvas;

	public void start() {
		app = this;
		mainCanvas = new LQuest();
		Display.setCurrent(mainCanvas);
		mainCanvas.timer.start();
	}
}
```

★新名称「影さんマイルド」。パソコンにやさしい影さん。しかし電池を交換するのが精一杯。（東京都足立区・**オカッチ20歳**）……
【**編**：実物の影さんも電池で動いているようなもんです。ほら，そろそろ電池の交換時期だ。**Dr.D**：今はこのままにしておこう。影：うっ。】

```
public void hantei() {
            //正解
            if(p==cur  && qtime>=0) {
                        ok=1;
                        score+=10;
                        hit++;
            }
            //不正解
            else {
                        ok=0;
                        hit=0;
                        if(score>=10)
                                    score-=10;
                        else  if(score>=0)
                                    score=0;
                        endck++;
                        //ゲームオーバー判定
                        //３回間違えると終わり
                        if(endck==3) {
                                    key=0;
                                    ok=-1;
                        }
            }
            q++;
            Init(1);
}

//画面描画
public void paint(Graphics g) {
            int[] selectcolor = {g.BLACK,g.RED};
            String[] kigou = {"+","-","*","/"};
            int x,y,i;

            //文字表示関連
            g.setOrigin(0,  0);
            g.lock();
            g.clearRect(0,  0,  getWidth(),  getHeight());

            //問題の設定
            if(setf==1) {
                        q_set();
                        setf=0;
            }
            //ゲームオーバー表示
            if(key==0) {
                g.setColor(g.getColorOfName(g.RED));
                g.drawString("GAME OVER",  5,  95);
            }
            //正解不正解コメント表示
            if(ok==1) {
                g.setColor(g.getColorOfName(g.LIME));
                g.drawString("Good!",  25,  95);
            }
            else  if(ok==0) {
                g.setColor(g.getColorOfName(g.RED));
                g.drawString("Bad!",  25,  95);
            }
            //ゲーム表示部分
            g.setColor(g.getColorOfName(g.BLUE));
            g.drawString(" L-Quest for i",  5,  10);

            g.setColor(g.getColorOfName(g.BLACK));
            g.drawString("Q:",  2,  25);
            g.drawString(Integer.toString(q),  12,  25);
            g.drawString("Score:",  30,  25);
            g.drawString(Integer.toString(score),  65,  25);
            g.drawString("Hit:",  30,40);
            g.drawString(Integer.toString(hit),  65,  40);
            g.drawString("Timer:",  30,55);
            g.drawString(Integer.toString(qtime),  65,  55);

            g.drawString(Integer.toString(n1),  10,  80);
            g.drawRect(25,70,10,10);
            g.drawString(Integer.toString(n2),  40,  80);
            g.drawString("=",  55,  80);
            g.drawString(Integer.toString(ans[p]),  65,  80);

            x=50;  y=110;
            g.drawString("ｷｺﾞｳ = ",  5,  y);
            for(i=0;i<4;i++) {
                g.setColor(g.getColorOfName(selectcolor[nowcolor[i]]));
                g.drawString(kigou[i],  x+(10*i),  y);
            }
            g.unlock(true);
}

//イベント処理
public void processEvent(int type,  int param) {
            switch (type) {
            case Display.KEY_RELEASED_EVENT:
                        //再始
                        if (param == Display.KEY_SOFT1)  {
                                    Init(0);
                        }
                        //Exit
                        if (param == Display.KEY_SOFT2)  {
                                    Main.app.terminate();
                        }
                        repaint();
                        break;
            case Display.KEY_PRESSED_EVENT:
                        switch (param) {
                        case Display.KEY_LEFT:
                                    if(key==1 && cur>0 && cur<=3) {
                                                nowcolor[cur]=0;
                                                cur--;
                                                nowcolor[cur]=1;
                                    }
                                    break;
                        case Display.KEY_RIGHT:
                                    if(key==1 && cur>=0 && cur<3) {
                                                nowcolor[cur]=0;
                                                cur++;
                                                nowcolor[cur]=1;
                                    }
                                    break;
                        case Display.KEY_SELECT:
                                    if(key==1)
                                                hantei();
                                    break;
                        }
                        repaint();
                        break;
            case Display.TIMER_EXPIRED_EVENT:
                        if(key==1) {
                                    qtime--;
                                    if(qtime<0) {
                                                hantei();
                                    }
                                    repaint();
                        }
                        break;
            }
}
}
```

《リスト3》「移植版　L-Quest for i」LQuest.jamのプログラム・リスト

```
PackageURL = LQuest.jar
AppSize = 2426
AppName = LQuest
AppVer = 1.0
AppParam =
AppClass = Main
SPsize = 0
LastModified = Tue, 13 Nov 2001 21:37:44
KvmVer =
```

★12月24日，僕の誕生日を迎えてしまいました。でもクリスマスイブと僕の誕生日が一緒だなんて…最悪です（涙）。（**福島県郡山市・近内沖尋19歳**）……**【つぐ美：あら，素敵じゃない！プレゼントも倍になるし，みんなに覚えてもらえるし，いいことばかりでしょ。】**

★★★★★★★ マイコンBASICマガジン ★★★★★★★

明日のスター・プログラマー

Programmer's Room

今月の投稿をズバリ解析！

編：ここは，今月の投稿プログラムの傾向やアドバイスを行う投稿者のためのコーナーです。さて，今月は……？

影：先月に引き続き，今月もちょっと元気のない投稿だったなぁ。もっと気合いの入った投稿作品を待ってるゾ。

編：とくに今月は，アイデア面では全然良さがなかったと言ってもいいですね。しっかり作品としての形にはなっているのですが，そこからの工夫が全然と言っていいほど見られません。遊び心が入ったプログラムがないという状況です。自作プログラムというのは，ある程度形になってからの工夫がおもしろいはず。ぜひとも，個性を持たせたプログラム作りに挑戦してほしいものです。

つぐ美：影さん，編さんからのちょっと辛口なコメントでしたが，こういうコメントこそしっかり受け止める必要がありますよ。さてDr.，辛口の評価が多いなか，今月の作品で目に付いたものは？

Dr.D：今月は，VBScript＋DirectXの「スピード」が目に付いた。ここまでしっかりしたゲームをVBScript＋DirectX環境で制作したのはなかなかじゃろう。じゃが，やはり定番ゲームなので，アイデア面ではマイナス・イメージじゃな。

編：HSP用の『ルパン』は，前述した遊び心がしっかり盛り込まれていて，おもしろい作品なのですが，やはりゲーム自体やキャラクター面でのオリジナリティに欠ける印象ですね。

影：アイデア面で唯一良かったと言えるのは，VB用の『Cut!』だけど，いかんせんこちらは，ゲームとしてはもう一歩完成していない感じだったな。

つぐ美：う～ん，みなさん厳しい意見が多いですね。そんな中，ベストに選んだ作品はどれなのでしょう？

Dr.D：今月は，実用プログラムのVisual C++用『BCJOYクラス＆サンプル』をベストに選ぼうと思う。今回はアイデア面でなく，実用性の高さを考慮しての選考じゃったな。

編：というわけで，今月のベスト・プログラマーは『BCJOYクラス＆サンプル』の作者，BlueCubeさんに決定です。おめでとうございます。

投稿の心得

影：さて，Dr.。先月は投稿に関する注意を説明してもらいましたが，今月は投稿への心得みたいなものを説明してもらえませんか？

Dr.D：うむ。一番大事なのは，投稿した作品は掲載される可能性があるということを意識することじゃ。掲載されれば多くの人が，プログラムを実行することになる。つまり他の人が実行してもなんら問題のない形にしておくことが大切じゃ。そのためには，誌面に掲載される説明文の充実，ファイルやOSの機能を使うプログラムであれば，自分以外の環境でも問題なく実行できるか，などをしっかりチェックしてから投稿してきてほしい。

もう一つ，掲載はWeb上でなく，あくまでベーマガの誌面にリストを含めて掲載されることを意識してほしい。とくに最近は，誌面に載るということを前提に考えていない投稿が多く目に付く。たとえば，説明文の不足や収録ファイルに依存した内容説明，リストの入力に関する説明不足，あまりにも多くの外部ファイルを使用するプログラム，といったところじゃな。

さらに，最近増えてきた画像ファイルなどを使用する作品も，できれば画像ファイルなしでも遊べる形にするか，画像ファイルを出力するリストを別に作成するなどの工夫がほしいところじゃ。

影：なるほど！投稿者は，このへんに注意をはらってみてください。

今月はもう一歩でした

●Visual Basic用『BASICSOFT NAVA 1.0』／柳瀬　峻（13歳）

つぐ美：数字でプログラミングするソフト。技術面は評価できるけど，やはりできることが限られてくる形式なのが残念でした。また，投稿の際には，超大作でもリストの添付をお願いしますね。

●Javaアプレット用『オセロ』／田楽（30歳）

影：定番ゲームそのままの形なので，やはりおもしろ味に欠ける。今回の「スタープログラマー」でのコメントをしっかり踏まえて作品作りに挑戦してほしい。

●Visual Basic用『アナログ時計VER. 1.01』／清家　巧（17歳）

編：機能が充実しましたが，やはりまだまだ中途半端な感じです。フォームの形を変更するなら，ぜひともリージョンは使ってほしいです。2001年2月号掲載のVB用『ATTACKER 2000』が参考になります。

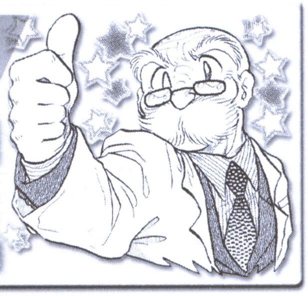
●Visual Basic用『五目並べの達人』／村田春佳＆瀧平恵介（17歳）

影：しっかり作られている五目並べだったけど，リストの長さが難点。超大作狙いであれば，もう少しこだわりがほしい。

投稿ありがとう

村田春佳＆瀧平恵介／石山明弘／三好利治／killsetaka／田楽／ロディマス／大島篤史／金井義男／赤羊旅団／原　崇史／はりまお／BlueCube／森　博行／小倉邦治／戸田文彰／黒田国男／清家巧／くろすけ／木村雄一／Mr.親父／白川智也／柳瀬　峻／折原　昇／三沢武志／たくちゃん／ヒウラ・トオル／マークン／梶浦徹矢／大熊通彦／龍夫
※全員紹介できなくてごめんなさい

Dr.Dのベスト・プログラム研究室



今月のベスト・プログラム：Visual C++用『BCJOYクラス&サンプル』（54ページ掲載）

今月のベスト・プログラムは，実用プログラムのVisualC++用『BCJOYクラス&サンプル』。アイデア面での目新しさはないが，本格的なゲーム・プログラムにはぜひとも取り入れたいジョイスティック処理を手軽に利用できるプログラムなので，実用性が高いと判断してのベスト選考じゃ。

今回は，このプログラムで使用されているAPIと構造体の解説と仕組み，キーボードへの対応，そしてボタン入力のチョットした改造を紹介してみよう。

(Dr.D)

使用APIの解説など

まず，このプログラムの特徴の一つに，クラスとして定義してある点が挙げられる。JavaやC++に触ったことのない読者には馴染みがないかもしれんが，今回のところは関数を含めて定義できる大きな構造体とでも思ってくれればOKじゃ(この説明は，あくまで簡易説明用の例えなので，鵜呑みにはしないように)。

さて，そのクラス内でジョイスティックを使用するための関数を定義しているわけじゃが，その定義に使用されている命令(API)が，“joy Get Dev Caps”と“joy Get Pos”じゃ。この二つの命令が，今回のプログラムの肝であり，この命令の動作を理解すれば，クラス内で定義されている関数の動きも理解できるはずじゃ。それでは，この二つの命令について説明していこう。

“joyGetDevCaps”命令

まずは，joyGetDevCapsの説明。この命令は，名前から推測できるようにジョイスティック・デバイスを得るための関数。いわば初期化処理を行うわけじゃ。この初期化が成功すると，ジョイスティックの情報が，この命令の2番目の引数に指定されているJOYCAPS構造体で宣言された変数に代入される。このJOYCAPS構造体は，ジョイスティックに関する値を非常に多く含んでおる構造体じゃ。興味のある読者は，ヘルプから検索して宣言内容を見てみると良いじゃろう。

“joyGetPos”命令

つぎはjoyGetPos命令の説明。この命令は，使用中のジョイスティックの情報を得る命令じゃ。この命令を実行すると，ジョイスティックの情報が2番目の引数で指定されているJOYINFO型の構造体で宣言されている変数に代入される。このJOYINFO構造体は比較的シンプルで，X方向，Y方向，Z方向，そしてボタンの情報を持つ四つの変数がある形じゃ。これらの変数の値から，指定されたボタンや現在押されている方向キーなどを知ることができる。そうした処理を行っているのが，buttonやdir関数じゃな。これらの関数で行っている処理は比較的単純なので，前述したように説明してきた二つの命令の動きがわかれば，仕組みを理解するのは，むずかしくない。

自分自身で，このクラスの仕組みを理解して，使いこなしてほしいものじゃな。

キーボードへの対応

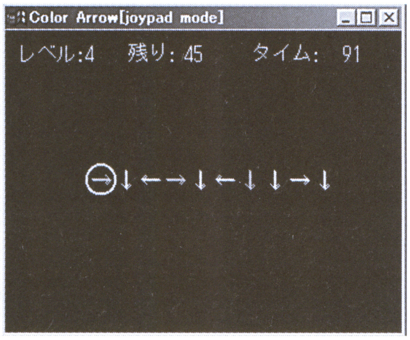
キーボード操作への対応も考えて関数を作っているのもこのクラスの特徴。じゃが，残念なことに掲載しているサンプル・ゲームのリストそのままでは，そのキーボード操作への対応記述に若干ミスがあり，対応していない。そこで，このページでキーボード対応への手順を説明していこう。

まず最初に，手順説明の前にどのようにして，キーボードへ対応しているかを簡単に解説しよう。このクラスでは，bool型(TrueかFalseの値を持つ)の変数modeが定義されており，このmodeの値によって，ジョイスティックであれば，前述したjoyGetPosを使用，キーボードであれば，GetKeyStateを使用することで，両方の操作に対応させている。このキーボード操作に扱う関数Get Key Stateに指定するキーコードは，クラス内で定義されている。

このキーボード操作をできるようにしたサンプルを**リスト1**に掲載してみた。このリストのように記述すれば，このプロ

《リスト1》キーボード操作対応への記述

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
	{
	(途中省略)
	//キーコードを好みで再定義したい場合は，配列にキーを指定しておく
	//この場合は，d[]が方向，b[]がボタンになる
	int d[4]={VK_DOWN,VK_LEFT,VK_RIGHT,VK_UP};
	int b[8]={'A','S','D','F','G','H','J','K'};

	(途中省略)
		case WM_KEYDOWN:
			if(wParam==VK_F1){
				joy->ChangeMode();
				mode=!mode;
			}
			if(mode){//joypadモード
			    SetWindowText(hWnd,"Color Arrow[joypad mode]");
			    joy->SetJoyMode();
			}
			else{//キーボードモード
			    SetWindowText(hWnd,"Color Arrow[keybord mode]");
			    joy->SetKeyMode();
				//以下２つでキーコードの再定義も可能
				//joy->KeySetDir(d,4);
				//joy->KeySetButton(b,8);
			}
			break;
```



▲Visual C++用「BCJOYクラス＆サンプル」

グラムのサンプル・ゲームがキーボードからでも遊べるようになるゾ。ただし，キーボードだけで遊ぶ場合は，サンプルならびにクラスのmode変数をfalseで初期化しておく必要がある。でないと，起動にジョイスティックを認識しようとして，場合によってはプログラムがフリーズしてしまうからじゃ。

BEST PROGRAM 改造してみよう

今回の改造は，CHECKER FLAGで影クンが述べているキートリガ設定を付ける，というものじゃ。このキートリガというのは，一度キーを押した後，そのキーを連続して判定するか否か，というものじゃな。

このキートリガ設定を，ジョイスティックの八つのボタンに対して行った改造プログラムを**リスト2**として掲載してみた。実現方法としては，各ボタンごとに状態を保存する変数(keyf[8])を新たに追加し，その変数の状態からすでにキーが押されている状態かを判断するようにした。この処理を行うには，キーがリリース(ボタンを離した状態)されたことを知る必要があるので，その処理を行う関数(bool buttonRelease(void);)もクラス内に追加した。

このリストを今回のサンプル・ゲームに適用すると，方向キーを押してからボタンを押すという操作になり，あまり適切ではない。また，このサンプル・ゲームで，より適切に実行させるにはbuttonC()関数も，keyfの値を参照するなどの変更をしないといけないのじゃが，今回の趣旨とは少しずれる内容になるので割愛させてもらう。

ゲーム内容によっては，このトリガ設定は有益になると思う。ぜひとも機会があれば使用してみてほしい。

《リスト2》ボタン改造の変更部分のリスト

```
bcjoyc.h部分の変更点
 ------------
 class BCJOY
           {
           private:
                        (途中省略)
                        //以下の変数を追加
                        int keyf[8];
 ------------
 BCJOY::BCJOY(int num)
            {
            int i;
            (途中省略)
            //追加した変数を初期化
            for(i=0;i<8;i++)
                     keyf[i]=0;
            }
 ------------
 bool BCJOY::button(byte n)
            {//以下のように記述
            if(n<1 && n>jc.wMaxButtons)
                     return false;
            if(keyf[n-1]==1)//キーフラグが立っている時は入力判定を行わない
                     return false;
            if(mode){
                     joyGetPos(jid,&ji);
                     if( ((ji.wButtons>>(n-1)) & 0x1)==1 ){
                              keyf[n-1]=1;//追加したキーフラグを立てる
                              return ( (ji.wButtons>>(n-1)) & 0x1 );
                     }
            }
            else{
                     if( ((GetKeyState(kb[n-1])>>15)&0x1)==1 ){
                              keyf[n-1]=1;//追加したキーフラグを立てる
                              return ( (GetKeyState(kb[n-1])>>15)&0x1);
                     }
            }
 }
 ------------
 //以下の関数を追加　クラス宣言部のpublic:にもプロトタイプを記述しておく
 bool BCJOY::buttonRelease(void){
            int n;
 for(n=1;n<=8;n++){
            if(mode)
                    {
                    joyGetPos(jid,&ji);
                    //ボタンが押されていないならkeyfを初期化
                    if( ((ji.wButtons>>(n-1)) & 0x1)==0 )
                             keyf[n-1]=0;
                    }
            else{
                    if( ((GetKeyState(kb[n-1])>>15) & 0x1)==0 )
                             keyf[n-1]=0;
            }
 }
            return true;
 }
 ------------
 main.cpp部分の変更点
 ------------
 void Scine2(HWND hWnd)
            {
            (途中省略)
            //if文の前にリリースを調べる
            joy->buttonRelease();
            if(joy->buttonC()==1 && joy->button(bc[c]+1) && a[joy->dir()-1]==bd[c])
 ------------
```

今月の掲載者から一言

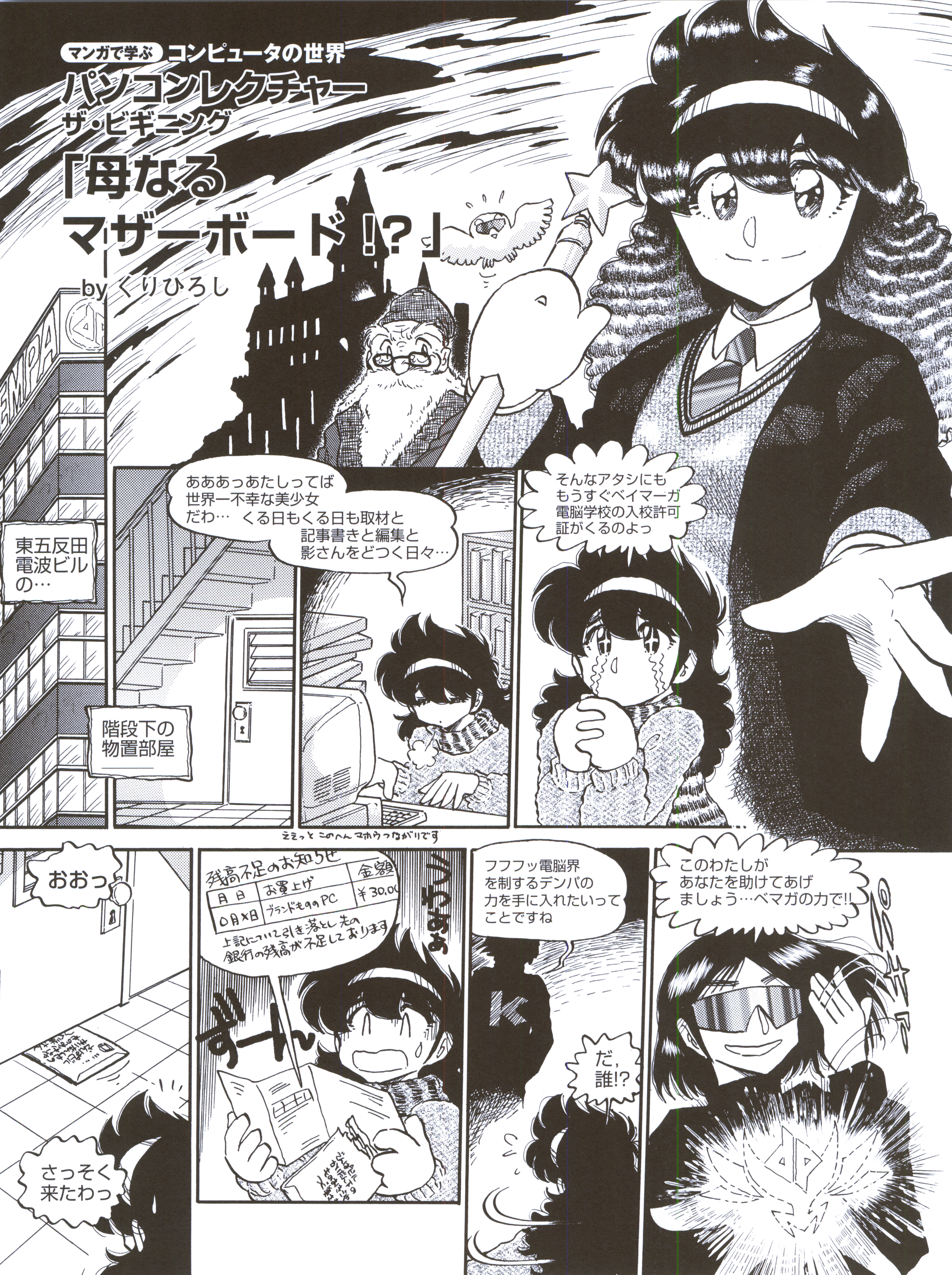
Windows（HSP V2.5）用「神経衰弱あわせてポン」三好　利治（46歳）

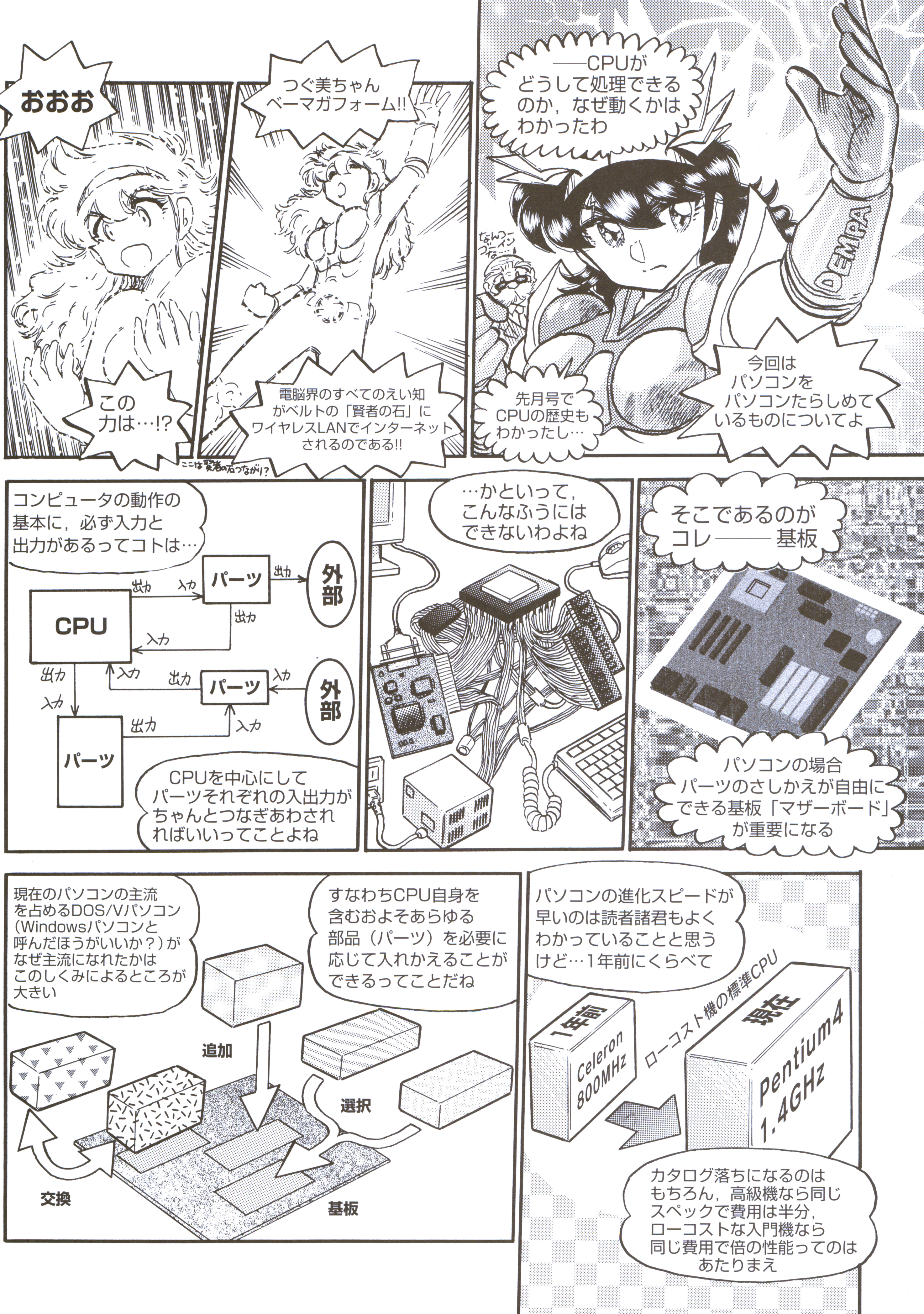
投稿コーナーのゲームは，とてもむずかしかったり（難易度が高い，歳だから？），すぐゲーム・オーバーになったりと，内容はすごいのに，おもしろくなかったりします。

また，ヘルプや説明書を読まないとわからないものが多いように思います。

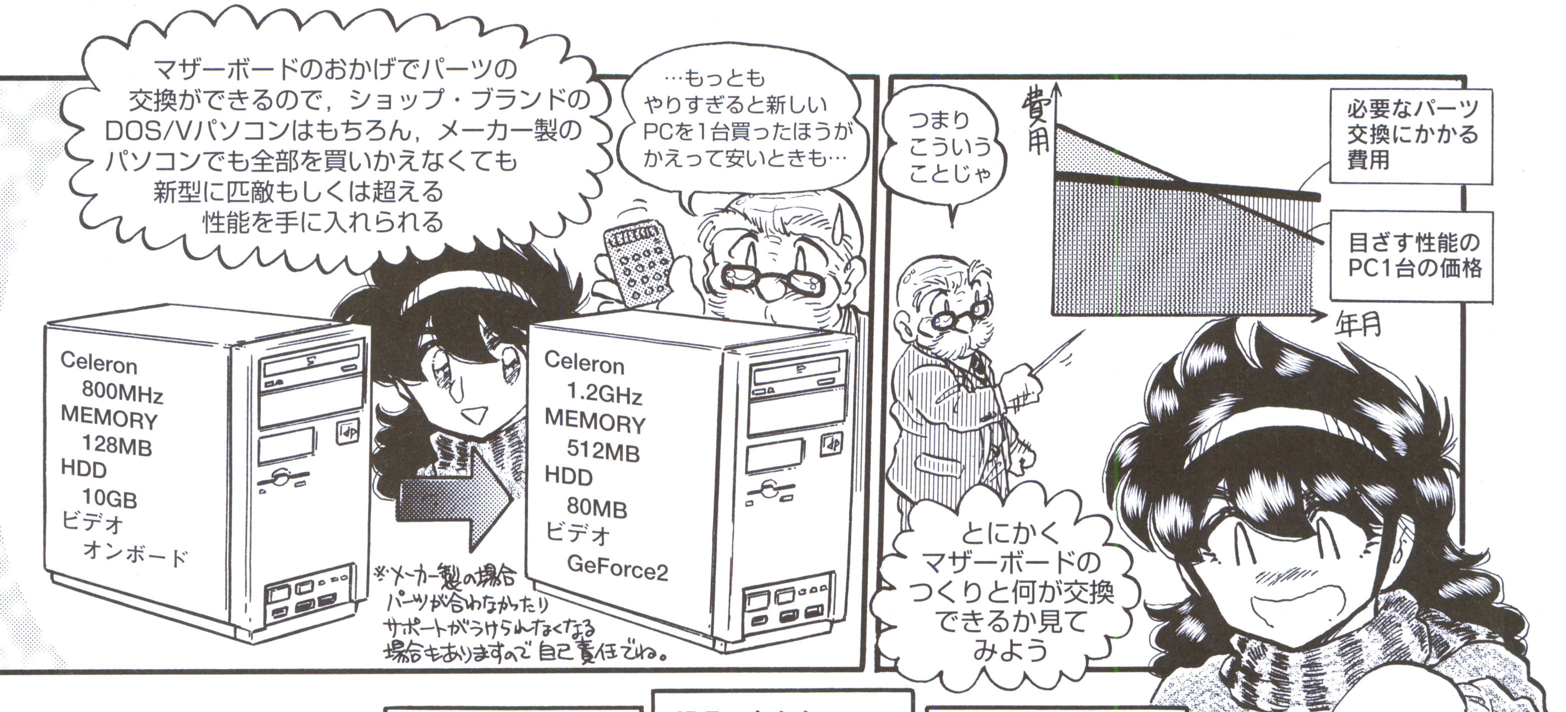
このゲーム（ゲームとは言えないか）は，アイデアも独自でないし，今の子にはまるで受けないでしょうが，家の者や子供は割り合いはまりました。

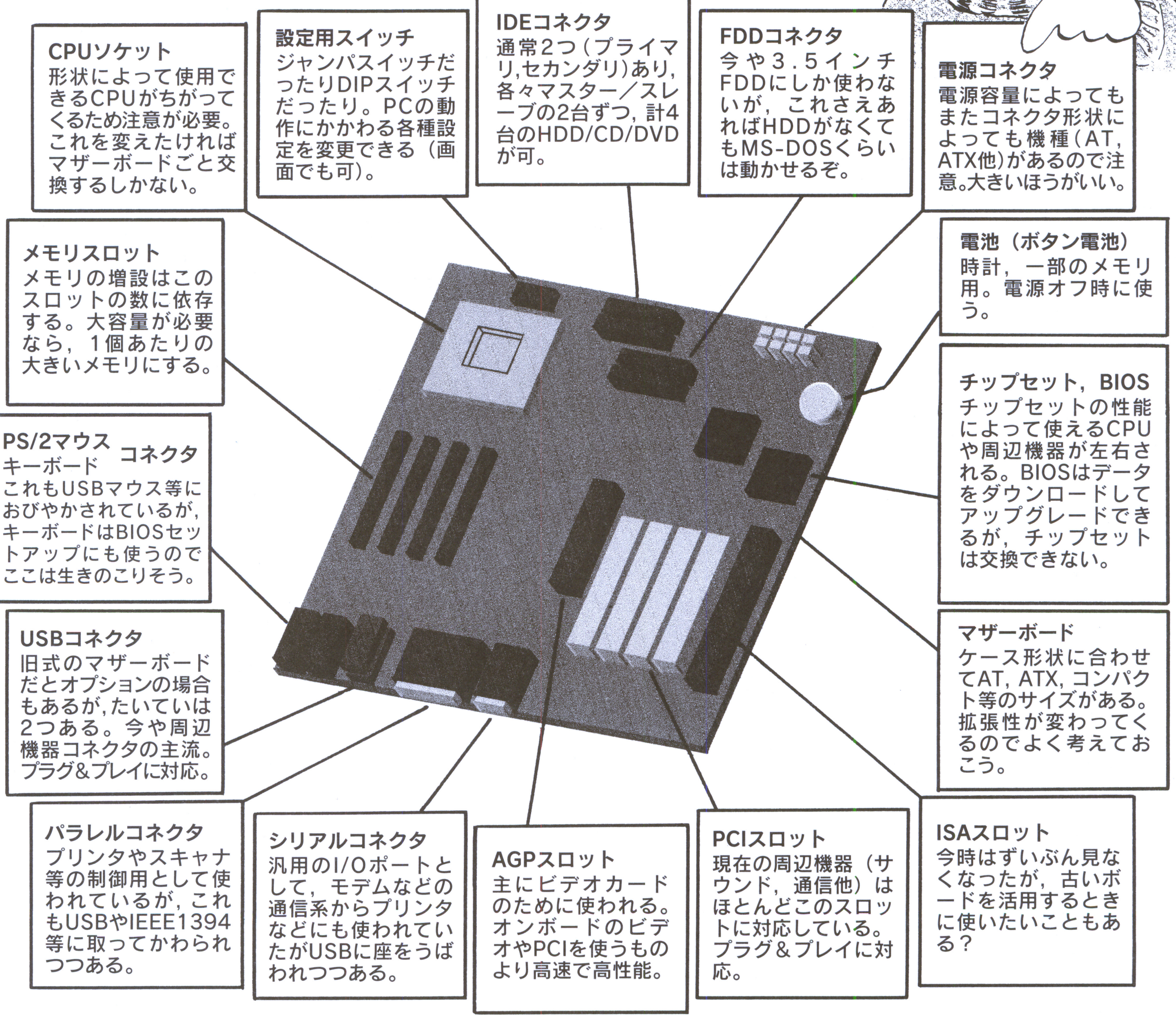
もし他に興味がありましたら，私の趣味サイトhttp://www.h3.dion.ne.jp/~yukomaru/index.htmへ一度どうぞ（チャウチャウとイラストがメインだったりしますが）。


マンガで学ぶ コンピュータの世界
パソコンレクチャー
ザ・ビギニング
「母なるマザーボード!?」
by くりひろし
そんなアタシにももうすぐベイマーガ電脳学校の入校許可証がくるのよっ
あああっあたしってば世界一不幸な美少女だわ… くる日もくる日も取材と記事書きと編集と影さんをどつく日々…
東五反田電波ビルの…
階段下の物置部屋
ええっと このへん マホウつながりです
このわたしがあなたを助けてあげましょう…ベマガの力で!!
フフフッ電脳界を制するデンパの力を手に入れたいってことですね
だ,誰!?
うわぁぁ
残高不足のお知らせ
上記について引き落とし先の銀行の残高が不足しております
ずーん
おおっ
さっそく来たわっ



おおお
この力は…!?
つぐ美ちゃんベーマガフォーム!!
電脳界のすべてのえい知がベルトの「賢者の石」にワイヤレスLANでインターネットされるのである!!
ここは賢者の石つながり?
――CPUがどうして処理できるのか，なぜ動くかはわかったわ
DEMPA
先月号でCPUの歴史もわかったし…
今回はパソコンをパソコンたらしめているものについてよ
コンピュータの動作の基本に，必ず入力と出力があるってコトは…
CPU
パーツ
パーツ
パーツ
外部
外部
出力
入力
CPUを中心にしてパーツそれぞれの入出力がちゃんとつなぎあわされればいいってことよね
…かといって，こんなふうにはできないわよね
そこであるのがコレ――基板
パソコンの場合パーツのさしかえが自由にできる基板「マザーボード」が重要になる
現在のパソコンの主流を占めるDOS/Vパソコン（Windowsパソコンと呼んだほうがいいか？）がなぜ主流になれたかはこのしくみによるところが大きい
すなわちCPU自身を含むおよそあらゆる部品（パーツ）を必要に応じて入れかえることができるってことだね
追加
選択
交換
基板
パソコンの進化スピードが早いのは読者諸君もよくわかっていることと思うけど…1年前にくらべて
ローコスト機の標準CPU
1年前
Celeron 800MHz
現在
Pentium4 1.4GHz
カタログ落ちになるのはもちろん，高級機なら同じスペックで費用は半分，ローコストな入門機なら同じ費用で倍の性能ってのはあたりまえ


マザーボードのおかげでパーツの交換ができるので，ショップ・ブランドのDOS/Vパソコンはもちろん，メーカー製のパソコンでも全部を買いかえなくても新型に匹敵もしくは超える性能を手に入れられる
…もっともやりすぎると新しいPCを1台買ったほうがかえって安いときも…
Celeron 800MHz MEMORY 128MB HDD 10GB ビデオ オンボード
Celeron 1.2GHz MEMORY 512MB HDD 80MB ビデオ GeForce2
※メーカー製の場合パーツが合わなかったりサポートがうけられなくなる場合もありますので自己責任でね。
つまりこういうことじゃ
費用
必要なパーツ交換にかかる費用
目ざす性能のPC1台の価格
年月
とにかくマザーボードのつくりと何が交換できるか見てみよう


CPUソケット
形状によって使用できるCPUがちがってくるため注意が必要。これを変えたければマザーボードごと交換するしかない。
設定用スイッチ
ジャンパスイッチだったりDIPスイッチだったり。PCの動作にかかわる各種設定を変更できる（画面でも可）。
IDEコネクタ
通常2つ（プライマリ,セカンダリ）あり，各々マスター／スレーブの2台ずつ，計4台のHDD/CD/DVDが可。
FDDコネクタ
今や3.5インチFDDにしか使わないが，これさえあればHDDがなくてもMS-DOSくらいは動かせるぞ。
電源コネクタ
電源容量によってもまたコネクタ形状によっても機種（AT，ATX他）があるので注意。大きいほうがいい。
メモリスロット
メモリの増設はこのスロットの数に依存する。大容量が必要なら，1個あたりの大きいメモリにする。
電池（ボタン電池）
時計，一部のメモリ用。電源オフ時に使う。
チップセット，BIOS
チップセットの性能によって使えるCPUや周辺機器が左右される。BIOSはデータをダウンロードしてアップグレードできるが，チップセットは交換できない。
PS/2マウス キーボード コネクタ
これもUSBマウス等におびやかされているが，キーボードはBIOSセットアップにも使うのでここは生きのこりそう。
USBコネクタ
旧式のマザーボードだとオプションの場合もあるが,たいていは2つある。今や周辺機器コネクタの主流。プラグ&プレイに対応。
マザーボード
ケース形状に合わせてAT, ATX, コンパクト等のサイズがある。拡張性が変わってくるのでよく考えておこう。
パラレルコネクタ
プリンタやスキャナ等の制御用として使われているが，これもUSBやIEEE1394等に取ってかわられつつある。
シリアルコネクタ
汎用のI/Oポートとして，モデムなどの通信系からプリンタなどにも使われていたがUSBに座をうばわれつつある。
AGPスロット
主にビデオカードのために使われる。オンボードのビデオやPCIを使うものより高速で高性能。
PCIスロット
現在の周辺機器（サウンド，通信他）はほとんどこのスロットに対応している。プラグ＆プレイに対応。
ISAスロット
今時はずいぶん見なくなったが，古いボードを活用するときに使いたいこともある？

さて大事なことを忘れないでほしいのだけれど…
パワーアップには限界があるよね？
おおっ
たとえば，CPUはソケットの形が最新型に変わったりメーカーがちがっても使えないことがあるから，マザーボードごと交換することになる
最新OS Windows XPでは，パーツをたくさん交換するとアクティベート（使用できるようにする手つづき）をやりなおさなきゃならん
うっ
うぬぬぬぬっ
ビデオカードの性能をいくら高くしても，CPUが遅すぎると性能を引きだしきれない――つまり一部だけパワーアップしてもダメなこともある
相性もあるぞ Pentium4の本当の実力を引きだしたければSDRAM対応のローコストでなく，高速のRDRAMを使わないと
そんなんできるわよ――1度にアップグレードするのは一つのパーツずつにするとか
パーツショップの人と良く相談するとか！
インターネットで情報を集めたりするとか…
何よりもベーマガ（本誌）を読めばいいのよォ!!
うわあぁぁ なんて手前ミソなんだあぁ
OUCH
す，すごいわっ これがベマガの力なのねっ…！
これでDr.Dにたよらなくても次なる周辺機器のしくみもわかりやすく…
フフフッそうかんたんにできるかな？
…ベーマガにDr.と呼べるのはワシ1人でいい…
え!? Dr!?
“電信”っ!!
げっ
ボリッ
いや，じつはボクも…
ASISTED BY Eric. Special thanks to Max.
2002
いいかげんにぃぃ しろおおおおおっ
あたし負けないっ
そ…そういえばチョコレートって…どうしたっけ…？
TO BE CONTINUED...
さあ！みんな！どこまでネタ元がわかったかな？
…でも このネタは多分つづかない♪

これから始めるブロードバンド活用術

ADSL HowTo!! 体験記

ADSLって何？ - ブロードバンド基礎講座 -

パソコンでＴＶ放送のように番組が楽しめたり，数10MBもの大容量ファイルのダウンロードが秒単位で行えるなど，インターネットのブロードバンド接続は，これからのパソコン活用にはもはや欠かせないものといえます。なかでも昨年から本格的に開始されたＡＤＳＬ接続は，導入費用も安くブロードバンドの超高速通信を手軽に実現できる魅力的な接続サービスです。そこで，ＡＤＳＬ入門者のためブロードバンド導入法を，筆者の実体験をもとに紹介しましょう。

ＴＶコマーシャルや雑誌で「ブロードバンド」という言葉を一度ならず目にするか聞いたことがあるはずです。実は，インターネットの世界をこれまでとは比較にならないほど広げてくれるのが，このブロードバンドなのです。

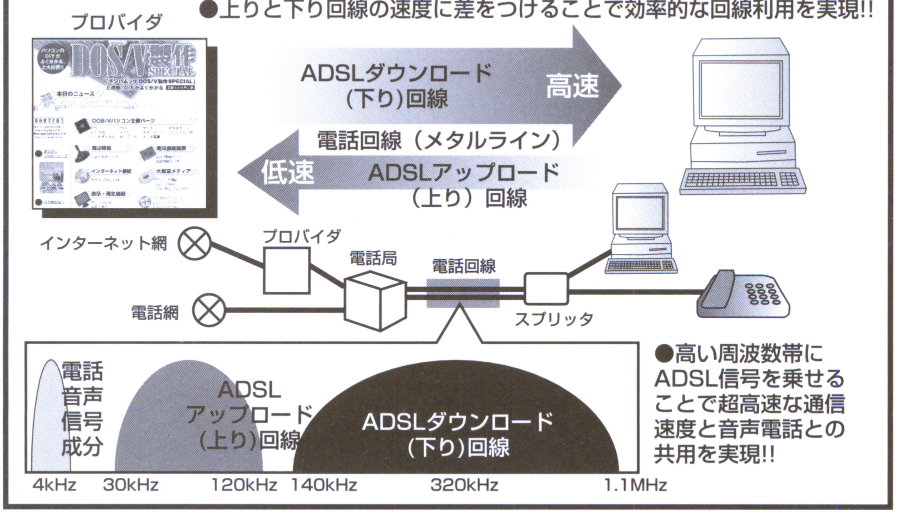
ブロードバンドを直訳すると，「広帯域」という意味になります。広帯域といわれてもピンとこないかもしれませんが，超高速なインターネット接続が可能になるデジタル回線といえば，わかりやすいかもしれません。インターネットへパソコンの内蔵モデムを介して接続している人も多いはずですが，画像や動画などをちりばめたホームページへアクセスすると，画面に表示されるまで，ずいぶん待たされることも珍しくありません。

これはパソコンとインターネット・プロバイダ（接続業者）の間を一般の電話回線を通して接続しているために，データの通信速度が遅くなりファイルサイズの大きな画像や動画データの転送に時間がかかることが原因です。

ところで，なぜ一般の電話では，通信速度が遅くなるのでしょう。細いパイプで相手側に水を流す場合を考えてみてください。一度に多量の水を流そうとするとパイプに入りきらずこぼれてしまいますが，少量の水であればこぼれずに相手側まで流すことができます。電話回線が細い（狭い）パイプで，データを水に当てはめると，データの一部分だけを相手に送ることができれば，確実に相手側に伝えることができるわけです。

このように電話回線では多量のデータを一度に送れないため少しずつデータを送る，すなわち遅い転送速度でしか通信が行えないのです。これを専門用語で「狭帯域」通信といいます。

パイプの直径を広くすれば，多量のデータを一度に送れ，高速通信が可能になるわけですが，これがブロードバンドと呼ばれる「広帯域」回線なのです。



《第１図》ADSLで高速通信が可能になる仕組み

ブロードバンド回線の種類とその特徴

ひとくちにブロードバンドといっても，さまざまな種類があります。一般のパソコン・ユーザーが導入できるものとしては，

- ●ＡＤＳＬ回線
- ●ＣＡＴＶインターネット回線

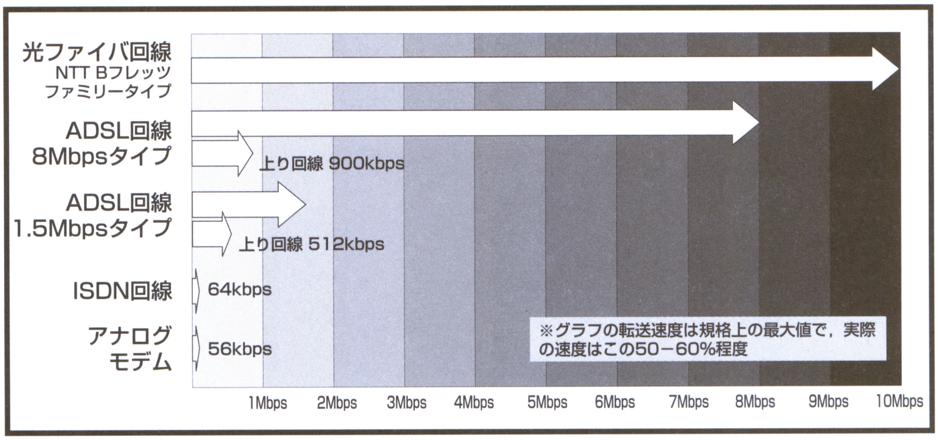
●光ファイバー回線
●無線インターネット回線
があります。

ＡＤＳＬは，現在最も注目を集めているブロードバンド回線で，一般の電話回線をそのまま利用できるため，導入にかかる工事費や回線使用料が，安く済むのが特徴です。

また，１本の回線で音声通話とデータ通信の２回線を同時に利用できるのも，ＡＤＳＬならではのメリットです。

ところで，低速な電話回線をどうしてブロードバンドとして利用できるのでしょうか？　音声とは格段に離れた極めて高い周波数帯域の信号と特殊な転送方式を利用することで，ＡＤＳＬは超高速なデータ通信を実現しているのです。

ＡＤＳＬは，「Asymmetric Digital Subscriber Line」を略した言葉で，非対称デジタル加入者回線のことです。この非対称というのは，ＡＤＳＬの特徴ともいえるもので，要するにデータのダウンロード（インターネット→パソコン／下り回線）とアップロード（パソコン→インターネット／上がり回線）で通信速度が異なる通信方式です。



《第２図》従来の接続方式とADSLの通信速度比較

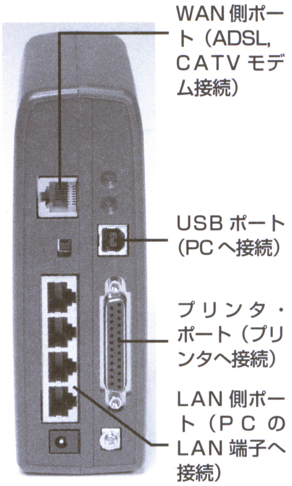
ＡＤＳＬは，ダウンロード側を高速化することで，１本のデータ回線を効率良く使い超高速なデータ通信を可能にしています。

ホームページにアクセスしたりストリーミングと呼ばれる動画や音声の配信サービスを受ける時は，パソコン側でデータを受信することがほとんどで，ダウンロードを高速化したＡＤＳＬは理にかなっているわけです。とはいえ，アップロードの転送速度も従来の接続方式よりもはるかに高速です。なお，ＡＤＳＬには，データ通信と音声通話を共用する「タイプ１」とデータ通信だけの「タイプ２」の２種類の回線があり，ＡＤＳＬ導入時にいずれかのタイプを選ばなくてはなりません。

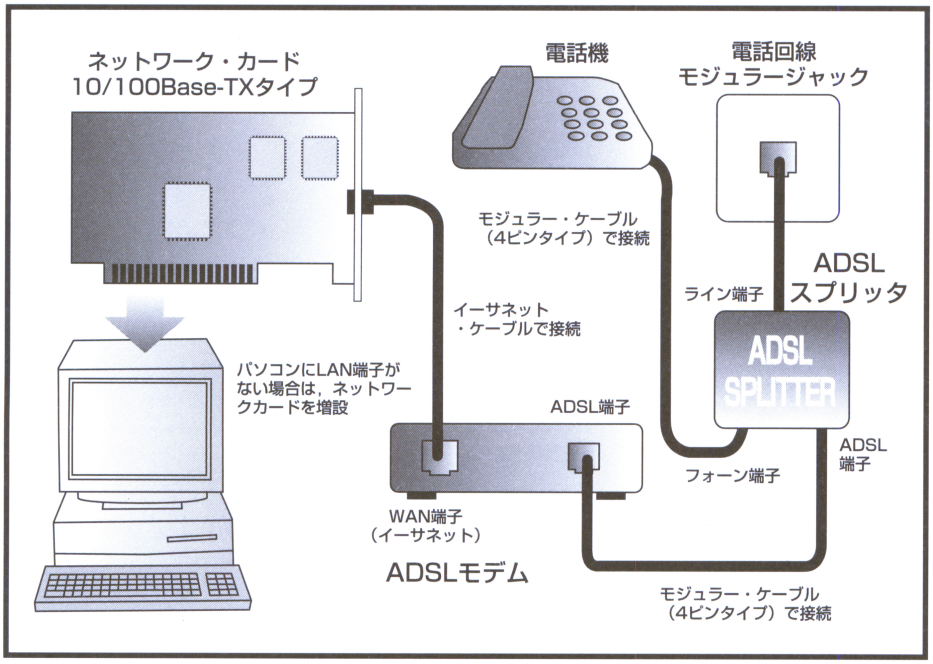
次のＣＡＴＶインターネット回線は，ケーブルＴＶ（有線ＴＶ放送）の回線をデータ通信に利用した方式です。ＡＤＳＬが登場する以前は，ブロードバンドの有力な候補に挙げられていましたが，ケーブルＴＶのサービスエリア内以外は

複数のパソコンからワイヤレスで高速インターネット 無線LAN対応ブロードバンド・ルータの新製品

ノートパソコンを中心に，ケーブル接続が不要な電波を使った無線ＬＡＮシステムが大人気だ。アイコムの「SR-11FB」（価格34,800円）は，無線ＬＡＮとインターネット・ルータを融合したワイヤレス・ブロードバンド・ルータの最新モデルで簡易プリンタ・サーバ機能もある優れもの。アマチュア無線や業務無線で定評のある同社の製品だけに，とくに信頼性の高い無線ＬＡＮ機能には注目。面倒なケーブル配線工事のいらないSR-11FBは，仕事場や家庭を問わずブロードバンド導入には，まさにＦＢ（ファインベター）な選択だろう。



別売の無線LAN PCカードをセットすれば，SR-11FBを通じてノートからワイヤレスでブロードバンド接続が行える



《第３図》ADSL 回線の接続方式（パソコン１台の場合）

利用できないなどの制約から，地方などでは普及が遅れています。

将来期待されているのが，光ファイバー回線（FTTH／Fiber To The Home）です。データ転送を光で行うことで，ＡＤＳＬやＣＡＴＶの数10～数100倍の極めて高速な通信が行えます。

ただし，高価な専用装置が必要で，工事に手間がかかることなどから，本格的な利用は数年程度はかかるようです。回線使用料が高価で，現時点では企業などのビジネス向けに限られています。

地域限定で一般家庭向けに低価格の接続サービスが始まっていますが，ＡＤＳＬとそれほど変わらない通信速度で，光ファイバー回線をあえて導入するする魅力はあまり感じられません。

回線工事が不要でＡＤＳＬ並の通信速度が得られるなど，メリットの多い無線インターネット回線ですが，現在行われているのはいずれも実験段階で，こちらも本格的な普及までもうすこし待たなくてはならないでしょう。

なお，店内などの限定された場所では，無線ＬＡＮネットワークを使ったブロードバンド接続サービスが各地で始まっています。

このように，ブロードバンドにはいくつかの方式があり，用途や環境に合わせて選択することができます。しかし，ブロードバンドを手軽に導入したいという場合は，ＡＤＳＬに勝るものはないでしょう。

ブロードバンドの最有力候補 ＡＤＳＬの性能は？

インターネットが，ブロードバンドでどこまで快適になるのか，気になるはずです。プロバイダが提供する回線のタイプで，同じ種類のブロードバンド回線でも，転送速度に大きな差があります。ＡＤＳＬの場合は，1.5Mと８Ｍ接続サービスが主流となっています。1.5M接続サービスは，ダウンロード時の通信速度が最大1.5Mbpsで，８Ｍ接続は８ Mbpsまでですが，実質のスピードはこの５割程度と思ってください。

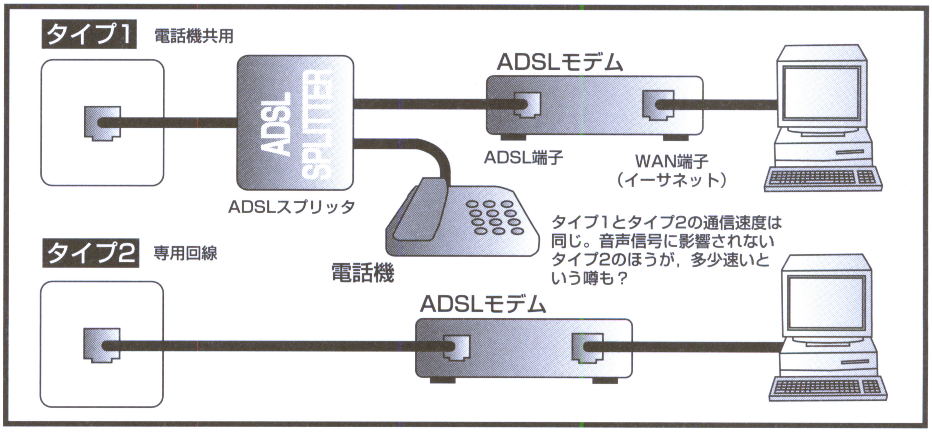
とはいうものの電話回線につなぐアナログモデムは56kbps，ISDN回線でも64kbpsですから，ＡＤＳＬの通信スピードの速さがわかってもらえるはずです。

たとえば，10MBのデータをダウンロードする場合を考えてみましょう。アナログモデムやISDN回線では，回線が混み合っていない時間帯でも20分程度はかかります。８Ｍ接続サービスのＡＤＳＬでは，実測値でわずか30秒程度で済んでしまいます。

分が秒単位になる恩恵は，ホームページへのアクセスにも明確な差を体験できます。これまで，「ズルズル」としか現れなかったホームページが，ＡＤＳＬではＴＶチャンネルを切り替えるように「パッパッ」と気持ちよく瞬時に表示してくれます。一度，体験したら元には戻れない快適なインターネット環境を手に入れることができるのです。

ダウンロードに比べ速度の遅いアップロードですが，今回導入した８ＭのＡＤＳＬ回線では，600kbps前後の速度が得られました。このスピードであれば，画像メールでもストレスなく送ることができるはずです。

ただし，ＡＤＳＬ回線を導入したといっても，断りもなくデータサイズの大きな画像をそのままメールに添付して送りつけるのは禁物です。相手がアナログモデムやISDN回線では，１通のメールを受け取るだけで，大変な時間がかかっ



《第４図》 2種類ある ADSL 回線のタイプ

てしまいますよ!!

ADSL導入の第一歩 必要な機材はこれだ!!

導入費用や回線使用料が安く済むとはいえ，ＡＤＳＬを利用するためには，これまでとは少々異なる機材が必要です。そのひとつが「スプリッタ」です。

ＡＤＳＬは，１本の電話回線で音声通話とデータ通信の両方が同時に行えることは先ほど紹介したとおりですが，音声とデータ通信の信号を識別し，二つのラインに振り分ける装置が，このスプリッタです。中身は帯域フィルタと呼ばれるパーツで構成されています。

「Phone」（フォーン）と「Line」（ライン），「modem」（モデム）の三つのモジュラー端子を持ち，フォーン端子は電話回線に，ライン端子は電話機，モデム端子はＡＤＳＬモデムへそれぞれ４ピンタイプのモジュラーケーブルで接続します。

フォーン端子の信号の中から低い周波数の音声と４kHz周辺にあるデータ信号をそれぞれ，ライン端子とモデム端子に分けるのが，スプリッタの役割です。

このスプリッタを介すことで，電話機で話しながら同時にインターネット接続を楽しむことができるのです。

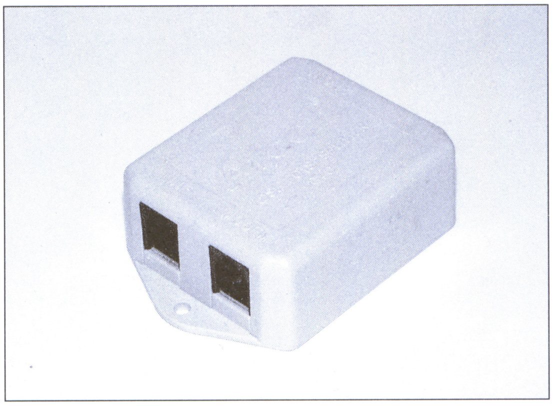
「ＡＤＳＬモデム」も，ＡＤＳＬ接続には欠かせない専用装置です。電話線を通るデータ信号をパソコンで扱えるデジタルデータに変換する点では，アナログモデムなどに似通っていますが，ＡＤＳＬモデムの場合，パソコンとの接続はイーサケーブルを用いるネットワーク接続が基本です。

ただし，パソコン間でファイルの共用などを行うＬＡＮ（ローカルエリア・ネットワーク）用ではなく，インターネット側のコンピュータとをつなぐ「ＷＡＮ」（ワイドエリア・ネットワーク）での接続となります。ＬＡＮやＷＡＮについては，次回以降に予定しているＡＤＳＬの実践的な導入方法で紹介しますので，ここでは，そういう言葉があるという程度にとどめておいてください。

なお，プロバイダが提供するＡＤＳＬ回線のタイプにより，利用できるＡＤＳＬモデムの種類が限られます。基本的には，プロバイダ専用モデムが必要になります。そのため，ＡＤＳＬモデムはプロバイダからの有料リース（貸出し）となっている場合が多いようです（場合によっては買い取りも可能）。

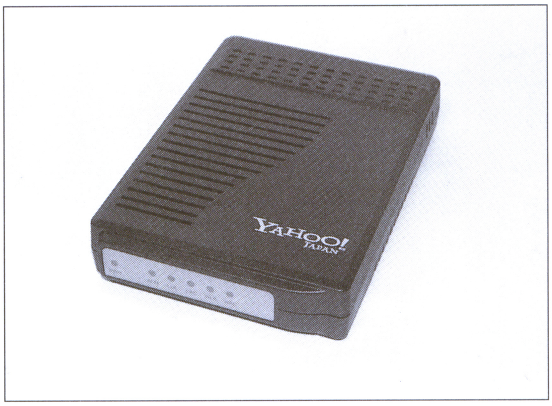
忘れてはならないのが，ＬＡＮ機能を搭載したパソコンか，パソコンへのネットワークカードの増設が必要になることです。このパソコンのネットワーク端子へＡＤＳＬモデムからのネットワークケーブルをつなぐわけです。ネットワーク接続に対応していれば，デスクトップ機やノートＰＣを問わず，ＡＤＳＬ回線が利用できるわけですが，端子が見あたらない場合でも，３千円程度で

用意する機材【ADSLモデム】



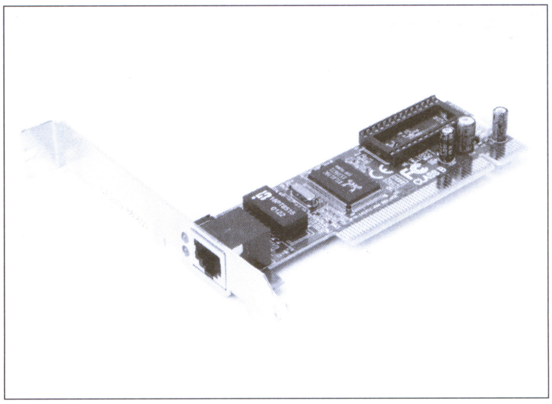
電話回線のラインから4kHz前後の低周波域の音声信号と，30kHzから1MHz周辺の高周波域を利用するＡＤＳＬデジタル信号を分離する，または音声信号とＡＤＳＬ信号を混合し電話回線に送る装置。通常はＡＤＳＬ回線導入時にプロバイダからＡＤＳＬモデムとともに貸与（有料）されることが場合が多いが，最近は市販品も出始めている。

用意する機材【スプリッタ】



ＡＤＳＬ通信のためのモデム装置で，信号の変調と復調を行う。スプリッタ側の電話ラインとパソコンへ接続するＬＡＮ（イーサネット）の仲介役にあたるのがＡＤＳＬモデム。プロバイダが提供するＡＤＳＬ回線は，いくつかの接続規格がありそれに適合した種類のモデムが必要。そのため，多くはスプリッタとともにプロバイダから有料リースする方式がとられている。写真は「Yahoo! BB」の専用ＡＤＳＬモデム

用意する機材【LANカード】



ＡＤＳＬをはじめブロードバンドの利用には，パソコンにＬＡＮ環境が必要だ。最近の市販パソコンの多くは，ＬＡＮ端子を標準装備しているが，旧モデルや自作パソコンでは，市販のＬＡＮカードでネットワーク機能を追加しなければなならない。このＬＡＮカードは，ＰＣショップなどで購入可能で，安価なものなら３千円前後で手に入る。購入時は「10／100Bese-TX」対応製品を選ぼう。

パソコンがインターネットＴＶに変身する ブロードバンド向けサイトにつなごう!!

ＡＤＳＬをはじめブロードバンド回線の普及とともに，その高速性能を生かせるインターネット・サイトが続々開設されているが，「ショートブレイク」（http://www.shortbreak.jp/）もその一つ。ショートフィルム（短編映画）の紹介をメインにしたストリーミング動画配信サイトで，一般公開されにくい珍しい自主映画なども楽しめるユニークなサイトだ。利用は有料だが，月500円（毎月８作品）を何度でも視聴可能で，映画ファンなら見逃せないはず。８ MbpsのＡＤＳＬ回線では，画質の劣化も少なく，ＴＶ画面で映画を見る感覚で楽しめる。ブロードバンドを導入したら，ぜひ一度アクセスしてみよう。

▲予告編を集めた無料視聴コーナーもあり，ブロードバンド回線の性能チェックにも使える

▲ストリーミング動画再生は，「Windows Media Player7」と「Real Player8」の両方式に対応

ネットワークカードが購入できますし，ノートＰＣであればネットワークＰＣカードをカードスロットに装着することで対応可能です。

複数のＰＣから同時利用 便利なブロードバンド・ルータ

以上のようにスプリッタ，ＡＤＳＬモデム，ネットワークカードの三つは，ＡＤＳＬ接続の三種の神器といってもよいでしょう。ただし，この場合は，１台のパソコンでＡＤＳＬ接続を行う環境でのことで，最近のように自宅や仕事先のように複数のパソコンからＡＤＳＬを利用する場合は，これらの機材に加えてブロードバンド・ルータと呼ばれる専用装置を用意することになります。

ルータとは，ネットワークのルーティング（道案内）を行う機器のことで，１本のインターネット回線を複数のパソコンで共用し，同時にインターネットへアクセスすることも可能です。要するにインターネット回線とパソコンのＬＡＮ環境の橋渡しするのが，ルータの役目と考えていただければよいでしょう。

それならば，ISDN回線でこれまで使っていたダイヤルアップルータがそのまま使えるのでは?! と思われるかも知れません。ブロードバンド・ルータとISDNダイヤルアップ・ルータは，働きとしては基本的に同じですが，対応するインターネット回線にISDNとADSLの違いがあります。そのため，ISDN用のルータをＡＤＳＬ接続に流用することは，一部の兼用機を除き，残念ながら不可能です。

ですが，ブロードバンド・ルータにはルーティングのほかにもいくつかの恩恵があります。IPアドレスを変換しＬＡＮ接続しているパソコンに仮想的に割り振り，数台のパソコンから同時にインターネットへアクセスできる機能（ＮＡＴ）や，一度ルータ側で設定しおけばパソコンで特別な設定を行わなくても，自動的にＬＡＮとインターネットの区別なく接続できるなど，便利な機能が使えるのもダイヤルアップ・ルータの特徴です。また，最近問題になっているハッカーの不正アクセスを防止するセキュリティ機器としても利用できます。

以前は高価でなかなか手を出せなかったブロードバンド・ルータですが，最近では１万円台のロープライス機種が続々登場しています。また，ＡＤＳＬモデムとルータが合体した多機能モデルも発売されています。パソコン１台だけでも利用可能なため，予算に余裕があるようなら，ADSLが初めての人にもブロードバンド・ルータの活用をお勧めします。

さて次回は，何かとトラブルの多いＡＤＳＬ回線の加入手続きと，ハードウエア関連の具体的なセッティング方法を失敗例を交えながら紹介します。

（文／福場龍夫）

用意する機材【ブロードバンド・ルータ】

複数のパソコンがある環境で，同時にインターネットへのブロードバンド接続が可能になる便利な機器が，このブロードバンド・ルータ。ＡＤＳＬモデム内蔵タイプや，無線ＬＡＮでのワイヤレス接続に対応した多機能モデルもあるが，基本的な機能に限定した製品であれば，１万円台から購入できる。セキュリティ機能もあり，パソコン１台から導入できるので，予算が許せば用意したい。大抵のブロードバンド・ルータは，ＡＤＳＬだけでなくＣＡＴＶインターネット回線にも対応している。

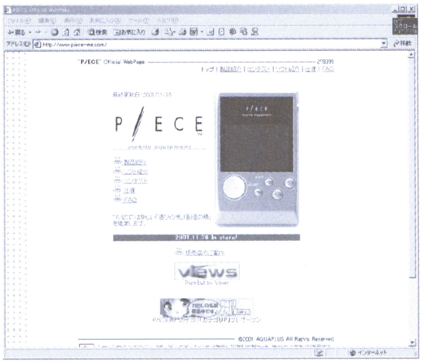
写真はメルコ製「BLR2-TX4」(価格13,800円)

PDAで自作ソフトを持ち歩く！

ポケットに入るPC用携帯ゲーム機「PIECE」プログラミング

PIECE本体
（これが原寸大だ）

101×65×17mm（縦×横×厚さ）
92g（電池含む）
発売元：アクアプラス
発売中
価格：9,800円（税別）
http：//www.piece-me.com/



ロイヤリティ＆ライセンスフリーで話題沸騰！　ユーザーオリジナルのソフトも続々登場！　話題に事欠かないPIECEでキミも自作ソフトを持ち歩こう！

（文／まかべひろし）

PIECEが発売してから早1か月。すでに各方面で「新しいプログラミングプラットフォーム」との話題を呼んでいますが，あなたはもうゲットしましたか？　そしてプログラミングしてますか？　付属ソフトもいろいろあるけれど，本誌的にはやっぱりプログラミングを楽しみたいところです。

さて，実際の製品版を見てびっくり。付属CDの中にアプリケーションやAPIリファレンス，マニュアル等が入っているのですが，まさかまさかPIECE自体の回路図が入っているとは。もしかしたら自作PIECEが作れるかも！？　さらにCDの中身を見ていくと…ソースの山。付属アプリのソースが見れるというのは事前に聞いてはいたのですが，カーネル部分のソースまでそのまま入っているのです。カーネルとはパソコンでいうBIOSやOSに当たる部分のこと。そんな中核部分まで見て改造できるなんて。あけっぴろげとも思える仕様公開ぶりに思わずドキドキしちゃいました。

さて，実際プログラミングするとなると，気になるのはその機能。どれくらいのことができるのかというのを調べてみました。

PIECEを使ってできること

●PIECEアプリ時

・普通にアプリを作れる。パッドで8方向入力可能。ボタンは押下情報のみで

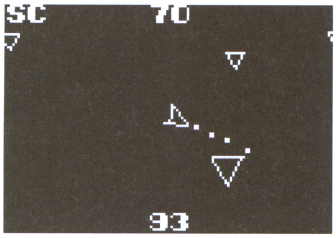
《第1表》PIECEのスペック表

CPU	EPSON S1C33209（32ビットRISC，24MHzで動作）
メモリ	SRAM 256Kbyte フラッシュRAM 512Kbyte
液晶	FSTN 白黒4階調 128×88ドット
音声出力	本体出力：モノラルスピーカ 外部出力：ヘッドホン端子
インターフェース	USB／赤外線
付属フォント	JIS第一・二水準漢字フォント（10×10ドット）
電源	単三電池×1
パソコン側環境	WindowsXP/2000/Me/98（要USB）

早くもユーザー・アプリ公開してます！

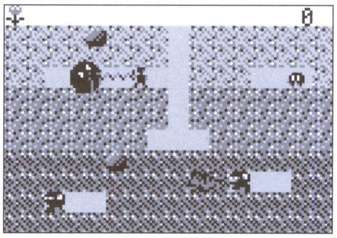
PIECEが発売されて，じつはその日から，ユーザーの手による新作アプリが続々と公開されており，その勢いには目を見張るばかり。
**　すでにメモリに入りきらないくらいのアプリがあり，どれを入れようか迷うほどです。そこで，数あるアプリの中からオススメのアプリを紹介しちゃいましょう。**

DeltaStar（ソース付）
BLACK FTZ氏制作
http://www2s.biglobe.ne.jp/~rokuhara/



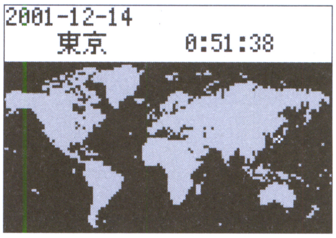
慣性のほどよく効いた滑らかな動きと，ワイヤフレーム描画によるシンプルなゲーム性が心地よいシューティング・ゲームです。操作はレーシング・ゲームよろしく，カーソルの上下で加速・減速。左右は自機を方向転換するのに使います。慣れて画面上を自由自在に飛び回るのは結構快感です。

BigBug
chaba氏制作
http://www4.synapse.ne.jp/chaba/piece/



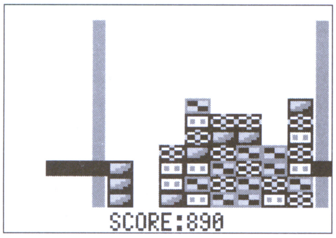
昔なつかしディ○ダグクローン的なアクション・ゲームです。ステージの中心にいるプレイヤーを上下左右に動かし，画面にはびこる敵を全滅させるのが目的。倒す方法は，なんと空気入れ。自転車のタイヤを膨らますあれです。敵を膨らまして足止めし，安全確認しつつさらに空気を入れて倒してしまいましょう。

WorldClock（ソース付）
DINNA氏制作
http://ueno.cool.ne.jp/ueno/6408/piece/



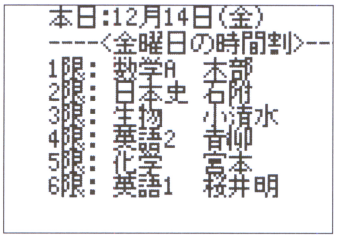
PIECEの電源である単三電池は海外でも買うことができます。ということは，海外旅行のお供にPIECEってのもアリ！　そこで，ほしくなるのはやっぱり世界時計ですよね。普段は普通の時計として使いつつ，海外に行ってもしっかり時刻を合わせて使えるとなれば，ぜひ入れておきたいソフトのひとつです。時間計算も，ソース付で参考になります。

PANEL Y/P
chaba氏制作
http://www4.synapse.ne.jp/chaba/piece/



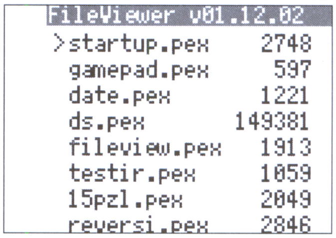
プレイヤーが操作するのはパネル本体ではなく，その横に表示されている1行のカーソル。これは自由に上下させることができます。で，左右キーを押すと，その行のパネルが1マス横にスライド。基本的な操作はこれだけで，同じパネルを4枚以上繋げ合わせて消していくのです。ルールも操作も簡単で，ちょっとした空き時間のプレイに最適です。

お手軽時間割表示ツール（ソース付）
KPCC氏制作
http://web12.freecom.ne.jp/~kpcc/



じつは結構有用？　とか思ってるのがこのツール。PIECEの中に週の時間割をセットしておいて，それを閲覧することができます。もちろん内蔵タイマによって表示する曜日が変わるので，いちいち曜日を合わせる必要もなし。ボクが学生の頃，時間割を忘れることが多々あったので，当時PIECEがあればすごく便利だったのでしょうね。

FileViewer（ソース付）
まかべひろし制作
http://www.marchen.to/~piece/



拙作でちょと恥ずかしいのですが，PIECEの中に入っているファイルを閲覧するのにぜひともほしいソフトです。というか，ほしいから作ったというのが本音です。これでテキスト・ファイルも持ち歩けますね。開発時にも便利なダンプ表示機能付。

アナログではない。
・PIECE内にファイル・システムあり。フラッシュメモリに容量が許す限りファイルを保存できる。読み出しのときは直接フラッシュメモリを参照可能。
・文字は漢字を第二水準レベルまで表示可能。半角のみだが拡大／縮小フォントも用意。
・赤外線でデータ送受信が可能。IrDAではないのでそれほど速くはない。
・タイマ内蔵。時刻をキーとしてアプリを起動させることができる。
・音声出力が可能。4チャンネル同時発生する。wave波形で16000Hz／16bit／モノラル相当なので，喋らせることも可能。ADPCMもサポート（カーネル・アップデート時）。
・スプライト機能搭載。8×8のBG背景での描画機能搭載。（カーネル・アップデート時）

●PC接続時
・PC側からPIECEのメモリを直接参照／書き換えが可能。ストレージみたいに使えるかも。
・PC側からPIECEのスピーカに対して波形出力が可能。これを使ってMP3再生も可能。
・USB接続を使ってPCにPIECE側からデータを送れる。

　ちなみにカーネル・アップデートによってやれることがどんどん増えているので，本誌発売時にはどうなっていることか予想もつきません。でも現状でもかなりおもしろいことがやれそうで，暇を見つけてはPIECEをいじくっています。

PIECEアプリの挙動について

作るときに覚えておきたいひとつ！ PIECEアプリは普通のC言語のようにmain ()からスタートしない，特殊な構成になっています。この概念がなかなかわかりにくいと思うので，軽くお話しておきましょう。

まず，リスト1のソースを見てください。これはまったく何もやらない，PIECE最小のプログラムです。

全部で三つの関数があります。最初に実行されるのはpceAppInit () です。ここで指定している pceAppSetProcPeriod (80)は，「pceAppInit () を抜けたあと，pceAppProc ()を80／1000秒間隔で呼び出すこと」という内容です。で，プログラムのメインはpceAppProc () で記述するというのがお約束事になっています。

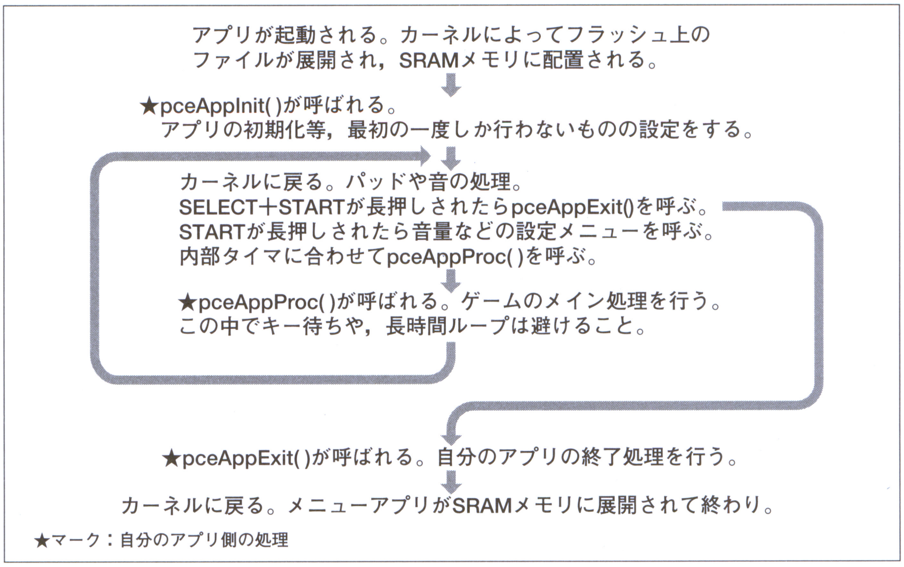
これはなぜでしょうか。それはPIECEの動作を考えると納得がいきます。基本的にCPUは，関数の途中で他のプログラムを割り込ませることができません。そういう前提の元でしょう，PIECEは第1図のような構成になっています。

この図を見ると，アプリの終了処理や設定メニューなどは，pceAppProc ()が呼ばれる合間に処理されることがわかります。また，その他パッドや音の処理なども行われているので，pceAppProc () の中でループを行ったりするのはご法度であるとわかるでしょう。これはPIECEに限ったことではなく，じつはWindowsやMacもこんな感じの処理を行っているのです。このように，定期的にOS側から呼ばれる関数をプロシージャと呼びます。

BASICなどの高級言語を使っているとなかなか普段はお目にかかれない考えかたですが，PIECEアプリを作るには必須の考えかたです。最初は慣れないと思いますが，覚えると別のプラットフォームでも使える考えかたなので，サンプルなどのソースを参照しつつ，がんばってみてください。

《リスト1》最小プログラム・ソース

```
#include <stdio.h>
#include <piece.h>

void pceAppInit( void ) {
            pceAppSetProcPeriod( 80 );
}

void pceAppProc( int cnt ) {
}


void pceAppExit( void ) {
}
```



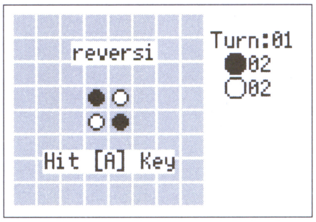
《第1図》PIECEの動作の流れ

リバーシを作ろうっ

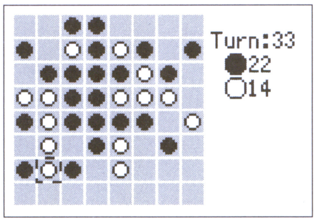
さてさて，今回のサンプル・ゲーム（リスト2）は，白と黒のコマがおなじみのリバーシ・ゲームです。上下左右でコマを置く場所を決め，Aボタンでコマを置きます。相手のコマを自分のコマではさむと，はさんだコマが自分のコマと同じ色になります。

で，全部埋まったときにコマの多いほうが勝ち，というゲームです。

ちょっとリストが長いですが，がんばって打ち込んでみてください。



◀サンプルのタイトル画面



◀サンプルのゲーム画面

ソフトウエア・コンテスト開催！

ユーザーが多彩なアプリを毎日のように発表しているほど盛り上がっているPIECEコミュニティですが，発売元のアクアプラス主催でソフトウエア・コンテストが開催されることが決まりました！ 見事入選すれば，雑誌やサイトで自作ソフトが発表され，しかも賞金まで入る滅多にないチャンス！

12月賞，1月賞，本選と3回にわけて開催され，この号が発売する頃には12月賞が決まってる頃だと思いますが，最終〆切は3月末なのでまだまだ時間はたっぷりあります。今から勉強しても遅くはないっ。アイデアを振り絞ってあっと驚くソフトを作ってみようじゃないですか！

【応募条件】
- フリーソフトとしてサイト，雑誌付録等に掲載可能であること
- 全年齢対象のもの
- 公序良俗に反するものは禁止
- プロアマ問わず
- 他者の著作権を侵害しないオリジナルのものであること（二次創作禁止）

【部門】
◆12月賞（2001年12月中の応募作品）
※すでに終了しました
◆1月賞（2002年1月中の応募作品）
◆本選・ゲーム部門
◆本選・実用ソフト部門

【受付期間】
12月賞……終了
1月賞……2002年1月1日～1月31日まで
本　選……2002年2月1日～3月31日まで

【賞の種類・賞金額】
12月賞 5万円 3本
1月賞 5万円 3本
本選・大賞（ゲーム・実用）100万円 各1本
本選・副賞（ゲーム・実用）50万円 各1本
本選・佳作（ゲーム・実用）30万円 各2本
本選・審査員特別賞 5万円 10本
本選・雑誌社賞 賞金は各雑誌社で決定

【発表時期】
12月賞……2002年1月予定
1月賞……2002年2月予定
本　選……2002年5月予定

【応募方法】
アクアプラス社宛てにメールもしくは郵送

くわしくはこちら
http：//www.piece-me.com/

《リスト2》サンプル・ゲーム「リバーシ」のプログラム・リスト

```
/* reversi, basicmagazine version. */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <piece.h>
#include <draw.h>

#define BOARD_X        4
#define BOARD_Y        4
#define STATUE_X       86
#define STATUE_Y       10

// 0: player, 1:cpu.
int nTurnTbl[2] = { 0, 1 };

int nTurnNum = 0;
int nTurnPlayer = 0;
int nCurX = 3, nCurY = 3;
int nChipCnt[3];
int nSkipCnt = 0;

// ゲームの進行カウンタ
int nGameScene = 0, nSceneCnt = 0;

unsigned char vbuff[ 128 * 88 ];
int nVramUpdate = 1;

// ボードのデータ
unsigned char byTbl[8][8];

// チップ画像
const unsigned char byChips[3][9*9] = {
        {
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
                1,1,1,1,1,1,1,1,1,
        } , {
                1,1,1,1,1,1,1,1,1,
                1,1,1,3,3,3,1,1,1,
                1,1,3,3,3,3,3,1,1,
                1,3,3,3,3,3,3,3,1,
                1,3,3,3,3,3,3,3,1,
                1,3,3,3,3,3,3,3,1,
                1,1,3,3,3,3,3,1,1,
                1,1,1,3,3,3,1,1,1,
                1,1,1,1,1,1,1,1,1,
        } , {
                1,1,1,1,1,1,1,1,1,
                1,1,1,3,3,3,1,1,1,
                1,1,3,0,0,0,3,1,1,
                1,3,0,0,0,0,0,3,1,
                1,3,0,0,0,0,0,3,1,
                1,3,0,0,0,0,0,3,1,
                1,1,3,0,0,0,3,1,1,
                1,1,1,3,3,3,1,1,1,
                1,1,1,1,1,1,1,1,1,
        }
};

// チップを1つ描画
void DrawChips( int x, int y ) {
        int i;
        for ( i = 0; i < 9; i++ ) {
                memcpy( &vbuff[ 128 * ( ( y * 10 ) + BOARD_Y + i ) +
BOARD_X + ( x * 10 ) ], &byChips[byTbl[y][x]][i*9], 9 );
        }
}

// ボード全体を描画
void DrawTable() {
        // 88*88 でボードを書く。1マス9*9, 1 line, 上下4pixel空け
        int x, y;
        for ( y = 0; y < 8; y++ ) {
                for ( x = 0; x < 8; x++ ) {
                        DrawChips( x, y );
                }
        }
}

// 画面内のチップ数をカウント
void CountChips() {
        int i = 0;
        nChipCnt[0] = 0;
        nChipCnt[1] = 0;
        nChipCnt[2] = 0;
        for ( i = 0; i < 64; i++ ) {
                nChipCnt[byTbl[i%8][i/8]]++;
        }
}

// 右上のステータス表示部
void DrawStatue() {
        char szBuf[100];
        CountChips();
        pceFontSetPos( STATUE_X, STATUE_Y );
        sprintf( szBuf, "Turn:%02d", nTurnNum + 1 ); pceFontPrintf(
szBuf );
        pceFontSetPos( STATUE_X, STATUE_Y + 10 );
        sprintf( szBuf, " ●%02d", nChipCnt[1] ); pceFontPrintf( szBuf
);
        pceFontSetPos( STATUE_X, STATUE_Y + 20 );
        sprintf( szBuf, " ○%02d", nChipCnt[2] ); pceFontPrintf( szBuf
);

}

// 回転カーソル描画
void DrawCursor( int nX, int nY, int nFlg ) {
        static int nStart = 0;
        int nWrite, i, j = nStart/2;
        for ( i = 0; i < 10; i++ ) {
                nWrite = 0; j++; if ( j % 8 < 4 && nFlg ) { nWrite = 3;
}
                vbuff[ 128 * ( ( nY * 10 ) + BOARD_Y - 1 ) + BOARD_X +
( nX * 10 ) - 1 + i ] = nWrite;
        }
        for ( i = 0; i < 10; i++ ) {
                nWrite = 0; j++; if ( j % 8 < 4 && nFlg ) { nWrite = 3;
}
                vbuff[ 128 * ( ( nY * 10 ) + BOARD_Y - 1 + i ) +
BOARD_X + ( nX * 10 ) - 1 + 10 ] = nWrite;
        }
        for ( i = 0; i < 10; i++ ) {
                nWrite = 0; j++; if ( j % 8 < 4 && nFlg ) { nWrite = 3;
}
                vbuff[ 128 * ( ( nY * 10 ) + BOARD_Y - 1 + 10 ) +
BOARD_X + ( nX * 10 ) - 1 + 10 - i ] = nWrite;
        }
        for ( i = 0; i < 10; i++ ) {
                nWrite = 0; j++; if ( j % 8 < 4 && nFlg ) { nWrite = 3;
}
                vbuff[ 128 * ( ( nY * 10 ) + BOARD_Y - 1 + 10 - i ) +
BOARD_X + ( nX * 10 ) - 1 ] = nWrite;
        }
        nStart++;
}

// その場に置けるかのチェック
// 返値：ひっくりかえせるコマの総数
int CheckPos( int nX, int nY, int nP ) {
        int nT[8][2] = {
                { -1, -1 }, {  0, -1 }, {  1, -1 },
                { -1,  0 },             {  1,  0 },
                { -1,  1 }, {  0,  1 }, {  1,  1 } };
        int nC = 0, nE = 1, i, j;
        if ( nP == 1 ) { nE = 2; }
        if ( byTbl[ nY ][ nX ] != 0 ) {
                return 0;
        }
        for ( i = 0; i < 8; i++ ) {
                if ( byTbl[ nY + nT[i][1] ][ nX + nT[i][0] ] == nE ) {
                        for ( j = 2; j < 8; j++ ) {
                                int nXX = nX + nT[i][0] * j;
                                int nYY = nY + nT[i][1] * j;
                                if ( nXX < 0 || 7 < nXX || nYY < 0 || 7
< nYY || byTbl[ nYY ][ nXX ] == 0 ) { break; }
                                if ( byTbl[ nYY ][ nXX ] == nP ) {
                                        nC += j - 1; break;
                                }
                        }
                }
        }
        return nC;
}

// 置いてひっくり返す処理
// 返値：ひっくりかえしたコマの総数
int ReversePos( int nX, int nY, int nP ) {
        int nT[8][2] = {
                { -1, -1 }, {  0, -1 }, {  1, -1 },
                { -1,  0 },             {  1,  0 },
                { -1,  1 }, {  0,  1 }, {  1,  1 } };
        int nE = 1, nC = 0, i, j, jj;
        if ( nP == 1 ) { nE = 2; }
        if ( CheckPos( nX, nY, nP ) == 0 ) { return 0; }
        byTbl[nY][nX] = nP;
        for ( i = 0; i < 8; i++ ) {
                if ( byTbl[ nY + nT[i][1] ][ nX + nT[i][0] ] == nE ) {
                        for ( j = 1; j < 8; j++ ) {
                                int nXX = nX + nT[i][0] * j;
                                int nYY = nY + nT[i][1] * j;
                                if ( nXX < 0 || 7 < nXX || nYY < 0 || 7
< nYY || byTbl[ nYY ][ nXX ] == 0 ) { break; }
                                if ( byTbl[ nYY ][ nXX ] == nP ) {
                                        for ( jj = 1; jj < j; jj++ ) {
                                                nXX = nX + nT[i][0] * jj;
                                                nYY = nY + nT[i][1] * jj;
                                                byTbl[ nYY ][ nXX ] = nP;
                                        }
                                        nC += j - 1; break;
                                }
                        }
                }
        }
        return nC;
}

// 全テーブルの初期化
void InitTable() {
        memset( byTbl, 0, 8*8 );
        byTbl[3][3] = 1; byTbl[3][4] = 2;
        byTbl[4][3] = 2; byTbl[4][4] = 1;
        nTurnNum = 0;
        nTurnPlayer = 0;
        nCurX = 3; nCurY = 3;
        nSkipCnt = 0;
}


//////////////////////////////////////////////////////////////////////
/////////

// PIECEのInit()
void pceAppInit( void ) {
        PCETIME strTime;
        pceLCDDispStop();
        pceLCDSetBuffer( vbuff );
        pceAppSetProcPeriod( 40 );
        memset( vbuff, 0, 128*88 );
        // random init.
        pceTimeGet( &strTime ); srand( strTime.s100 );
        pceLCDDispStart();
}

// PIECEのプロシージャ
void pceAppProc( int cnt ) {
        unsigned long lKey;
        int i, nBackupScene = nGameScene;
        switch( nGameScene ) {
        ////////////////////////////////////////////////////////////
        case 0: // タイトル
                if ( nSceneCnt == 0 ) {
                        memset( vbuff, 0, 128*88 );
                        InitTable();
                        DrawTable();
                        DrawStatue();
                        pceFontSetPos( 28, 15 );
                        pceFontPrintf( "reversi" );
                        pceFontSetPos( 16, 60 );
                        pceFontPrintf( "Hit [A] Key" );
                        nVramUpdate = 1;
                }
                lKey = pcePadGet();
                if ( lKey & TRG_A ) {
                        memset( vbuff, 0, 128*88 );
                        DrawStatue();
                        DrawTable();
                        nGameScene = 10;
                }
                break;
        ////////////////////////////////////////////////////////////
        case 10: // ゲーム
                for ( i = 0; i < 64; i++ ) {
                        if ( CheckPos( i % 8, i / 8, nTurnPlayer + 1 )
!= 0 ) { break; }
                }
                if ( i != 64 && nTurnNum < 64-4 ) {
                        int nPut = 0;
                        nSkipCnt = 0;
                        if ( nTurnTbl[nTurnPlayer] < 1 ) {
                                // player.
                                DrawCursor( nCurX, nCurY, 0 );
                                lKey = pcePadGet();
                                if ( lKey & TRG_UP ) { nCurY--; } if (
lKey & TRG_DN ) { nCurY++; }
                                if ( lKey & TRG_LF ) { nCurX--; } if (
lKey & TRG_RI ) { nCurX++; }
                                if ( nCurY < 0 ) { nCurY = 7; } if (
nCurY > 7 ) { nCurY = 0; }
                                if ( nCurX < 0 ) { nCurX = 7; } if (
nCurX > 7 ) { nCurX = 0; }
                                if ( lKey & TRG_A ) {
                                        nPut = 1;
                                } else {
                                        DrawCursor( nCurX, nCurY, 1 );
                                }
                        } else {
                                // cpu.
                                switch( nTurnTbl[nTurnPlayer] ) {
                                        case 1:
                                        {       // random.
                                                while( 1 ) {
                                                  nCurX = rand() % 8;
nCurY = rand() % 8;
                                                  if ( CheckPos( nCurX,
nCurY, nTurnPlayer + 1 ) != 0 ) { break; }
                                                }
                                                nPut = 1;
                                                break;
                                        }
                                }
                        }
                        if ( nPut ) {
                                if ( CheckPos( nCurX, nCurY, nTurnPlayer
+ 1 ) != 0 ) {
                                        ReversePos( nCurX, nCurY,
nTurnPlayer + 1 );
                                        nTurnNum++;
                                        DrawTable();
                                        DrawStatue();
                                        nTurnPlayer = ( nTurnPlayer + 1 )
% 2;
                                }
                        }
                        nVramUpdate = 1;
                } else if ( i == 64 && nTurnNum < 64-4 ) {
                        nSkipCnt++;
                        nTurnPlayer = ( nTurnPlayer + 1 ) % 2;
                        if ( nSkipCnt > 1 ) {
                                nGameScene = 20;
                        }
                } else {
                        nGameScene = 20;
                }
                break;
        ////////////////////////////////////////////////////////////
        case 20: // 勝敗表示
                lKey = pcePadGet();
                if ( nSceneCnt == 0 ) {
                        char szTbl[][16] = {
                                "Player", "COM"
                        };
                        char szBuf[100];
                        if ( nChipCnt[1] != nChipCnt[2] ) {
                                pceFontSetPos( 16, 31 );
                                pceFontPrintf( " Winner is, " );
                                pceFontSetPos( 16, 41 );
                                if ( nChipCnt[1] > nChipCnt[2] ) {
                                        sprintf( szBuf," ●%s ",
szTbl[nTurnTbl[0]] );
                                } else {
                                        sprintf( szBuf," ○%s ",
szTbl[nTurnTbl[1]] );
                                }
                                pceFontPrintf( szBuf );
                        } else {
                                pceFontSetPos( 16, 39 );
                                pceFontPrintf( " Draw Game " );
                        }
                        nVramUpdate = 1;
                }
                if ( lKey & TRG_A ) {
                        nGameScene = 0;
                }
                break;
        }
        // 画面表示
        if ( nVramUpdate ) {
                pceLCDTrans();
                nVramUpdate = 0;
        }
        if ( nBackupScene != nGameScene ) {
                nSceneCnt = 0;
        } else {
                nSceneCnt++;
        }
}

// PIECEのAppExit()
void pceAppExit( void ) {
}
```

もっと身近にプログラミング

チャレンジ!!

オリジナル・プログラム言語の制作！

その1 得手不得手を知って仕様を決めよう！

文 まかべひろし

最近PDAプログラムに夢中です。自分のプログラムを持ち運んで，すぐに人に見せられるのって楽しいですよー。やっぱモノ作りは，相手の反応が見られなくっちゃ，とか思ったり。そんなわけで近々，この楽しみを記事にしようと動いています。ふふふっ。

絵 桜部のとす

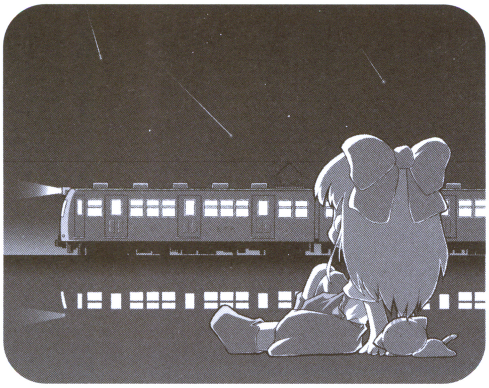
しし座流星群，キレイでしたね～って，今となっては昨年の話ですね。なにはともあれ，流れ星描いてみたくなっておもわず今回のカットで描いてみました。暗めなので印刷キレイに出るかな…。

catlikeホームページ ▶ http://catlike.marchen.to/

心機一転，今度はオリジナル・プログラム言語を作るという途方もない企画で，アルゴリズム（プログラムを作るときに使われる問題解決の方法や考え）を勉強していくのですよー！

2002年明けておめでとう～！

「戦」の年も明けて気が付いてみれば2002年。21世紀も，もう1/100が過ぎたわけですよ。でも，はやーい，なんて感嘆の声をあげている場合じゃありません。新しい年には新しいコーナーを，ということで，「もっと身近にプログラミング」もちょっとだけコーナー名を変えてみました！　でも中身はほとんど変わってないのは秘密です。



名前は変わった，で，何やるのさ？

今まで「もっと身近にプログラミング」ではいろいろなゲームの話題を取り上げて，その中で使われる主要なアルゴリズムについてのお話をしてきました。できるだけ言語というものに制約されることのないように，プログラムの考えかた，データの持ちかた，というところに主眼を置いて話を進めてきたのです。

で，そんなわけで今までずーっと言語の特性には触れないようにしてきたのですが，今回のお話は，もう思いっきり「プログラム言語」というものに触れていきたいと思います。プログラム言語にべったり。「えー。サンプルC言語で書かれてるけど，内容もC言語べったりになっちゃうの？」なんて心配ご無用。それもそのはず，今回のお話は，自分独自の「オリジナルなプログラム言語」を作ってしまおう，という内容なのです。

自分で言語なんて作れちゃうの？　って思っちゃうかもしれません。でも，作ってみれば意外に簡単。しかもその道は奥深く，手を加えれば加えるほど完成度が高くなっていくという，プログラムの分野として見ても，非常におもしろいも

のだと言えます。

本誌で連載がはじまった「HSP」も，そんなオリジナル言語の一つ。この連載を見てがんばれば，もしかしたらHSPを越える言語を作れちゃうかもしれませんよ！??

じつは，catlikeのソフトもほぼすべて，同じ「オリジナル・プログラム言語」のシステムで作られていたりして。「めるへんうぉーかー」シリーズは毎回ゲーム内容ががらりと変わっていますが，実行ファイル自体は同じ言語システムを使い，その都度必要な機能をバージョン・アップさせています。ですから，一番古い「ポーカー」の実行ファイルに，一番新しい「セブンブリッジ」の実行ファイルを上書きしても「ポーカー」が遊べてしまいます。これは，言語システム上で動いているソフトという特性が実現させる妙と言えるでしょう。

catlikeのソフトでは，そんな言語システムの上に，独自の言語で書かれたテキスト・ファイルが存在します。テキスト・ファイルには，どんなふうに独自の言語が書かれているかと言うと，

```
// 1秒ウェイト
#FUNC_TimeWait
:$TIME=$_TIME[0]
#_FUNC_TIMEWAIT_LOOP
:IF($_TIME[0]<$TIME+60){:GO'_FUNC_TIMEWAIT_LOOP'}
:RT
```

▲オリジナル言語，こうしたい！　みたいなイメージです。

こんな感じです。見たことのない言語ですので，内容が理解できないのは当然。1秒ウェイトを行うプログラムのつもりです。一応，BASICをもとにした言語形態にしてありますから，なんとなく意味はわかるかもしれませんね。

ゲームのプログラムは，こんな感じでテキスト・ファイルにその進行内容を記述していって，言語システムはそれぞれの命令内容を処理しています。このように，独自言語で書かれたプログラムを「マクロ」や「スクリプト」と呼ぶこともあります。言語システムは，このマクロをひとつ一つ解析しながら処理していきます。このような方法を，インタプリタ形式と呼びます。BASICでも採用されてる方式だったりするのですが，ここでは割愛。

しかしなぜ，わざわざプログラムを「言語システム」と，その上で動く「マクロ」の二つにわけて作っているのでしょう。このようにするからには，何らかの利点があるはずです。もちろん，このような2重構成にしたことによる欠点も生じます。そこでまずは，今回は導入部ですから，言語システムを作ることによる利点，欠点について見てみましょう。

オリジナル言語！利点と欠点を考える

まずは，言語システムを作る利点です。作ると何かしら良いことがある，って思えなければ，わざわざ言語システムなんて作りませんよね。名称や構造がかっこいいから，なんていうのは，プログラマーのエゴにすぎないのです。明確な利点を，考えてみましょう。

利点

❶ 機能単位でプログラムをするため，プログラムが単純

プログラマーが一番喜べる点だと思います。たとえば，入力機器一つ取っても，キーボードやマウス，ジョイパッドなどがあります。また，ゲーム・プログラムの場合，この入力に対応してキャラクターが動いたり，ゲームが進行したりなど，複数の機能が混在したプログラムを組まなくてはいけません。言語システムでは，それらを機能別に言語命令として作っていくため，それぞれが組み合わさってどう動くか，ということをさほど考えずにすみます。組み合わせの部分は，後ほどのマクロ記述の仕事。キーボードの押された状態を得る機能がほしければ，それだけを行う命令を作ればいいのです。明確に機能別になるため，言語システムのプログラムは単純になります。

❷ プログラムの面倒な部分を自分の好みで隠蔽することができる

これはC言語の関数やBASICのサブ・ルーチンを作ることに酷似しています。ゲームを作るときなど，たとえば「文字を表示する」とか「CGを読み込む」などのサブ・ルーチンを用意する人は多いと思います。サブ・ルーチンを使わないと，意外にプログラムとは面倒になるものです。もちろん，同じような処理を何度も行うときに有用なのがサブ・ルーチンなのですが，たとえば文字表示する前にはフォントを用意しなきゃいけない，使い終わったらフォントを削除しなきゃいけないなど，面倒な処理をルーチン内で行うようにするものです。

言語システムを作るとき，自分の好みでこれらをすべて隠ぺいし，マクロ上からは見えなくすることができます。命令や機能はすべてサブ・ルーチンであり，マクロ上ではそのサブ・ルーチンをどの順番で，どういうパラメータを与えて呼び出すのかを書くもの，とも言えます。

❸ 一度作ってしまえばデバッグが楽

デバッグに関しては，ソフトの信頼性を上げるのに役立ちます。言語システムは別にただ1本のソフトのためにつくるものではありません。同じ言語システムを使いまわし，いくつものソフトを作り出すことが可能です。足りない機能があれば，命令として言語システムに追加することで，どんどんシステム自体を育てていくことができます。

つまり，一度完成してしまえば新規機能として追加する部分のみ集中的にデバッグすればよいので，バグが出る可能性を押さえることができるわけです。

❹ マクロの移植性が高い

この利点は非常に大きいものと言えます。Windows版とMac版の言語システムを用意したとしましょう。言語システム自体はもちろん別のプログラムとなりますが，そのシステムが解析するマクロ側がまったく同じだった場合，Windows用ゲームもMac用ゲームも，一つのマクロで動いてしまいます。

外部にシナリオを用意するだけの単純なアドベンチャー・ゲームなら利点と呼べるほどではありませんが，マクロ言語だけでゲーム・システムが構築できるとなれば話は変わります。

そして，同じようにPlayStation版，Xbox版と用意することで，ひとつのマクロ・プログラムを多くのプラットフォームで動作させることが可能なのです。

❺ プログラマー以外の人がプログラムに参加できる

これ，ホントに利点なの？　と思っちゃいますか？　自分以外にプログラマーは必要ない！　というプログラマーさんが世間に多いのも事実です。ですが，たとえばシナリオ・ライターの人がプログラムに参加できる環境を作ることによって，ゲームの演出部分等を他の人に任せ，プログラマーは言語システムの安定化や機能アップに注力することができます。

そして，プログラマーの負担が軽減するということは，予算の効率化，ひいては経費削減につながるわけで！　…すいません，この原稿，マイコンBASICマガジンのプログラム講座用でしたね。

市販ソフトの世界では，いかにお金を効率よく使うかという論争が年がら年中繰り広げられていたりして，言語システムってのは，それに一役買ってたりするんですよ。お金の話はともかく，プログラマーも自分の負担を減らせるなら，結構喜ぶ人も多いんじゃないかと思います。

欠点

❶ 言語解析ルーチンを作るまでが手間

プログラムを気軽に考えている人には結構深刻な欠点です。何せ，最初の設計をするところから，画面に何か結果が出るまでが，すさまじく長い。せめて文字のひとつでも出ないことには，プログラムを組んでいておもしろくもなんともないですよね。

まぁ，この問題は，連載を追っていけば自然に解決できるようにするつもりですので，これについては考えないでおきましょう。

❷ 動作速度が遅い

この「言語システム」の最大の欠点です。マクロ・プログラムを解析しながら実行していくので，どうしても普通にプログラムを組むよりも動作速度が遅くなってしまいます。工夫しだいで，ある程度の高速化は図れますが，やはり手順の都合上，どうあがいても直接プログラムするより遅くなるのは否めません。

❸ 最初に許容した範囲内でプログラムを組むので，大規模な変更ができない

この問題は，同じシステムを長く使えば使うほど，大きな問題として，のしかかってきます。言語システムの限界は，命令単位の機能強化がメインであって，言語システムの根本を変更することができない点にあります。

この場合には，言語システムを新たに設計しなおすのが解決法となりますが，大規模な変更を行いたくなる頃には，度重なる機能強化でシステム自体も複雑になっていることが多かったりするので，素直に設計しなおすのが良い方法だと言えましょう。最近は技術の進化が早いですので，ひとつのシステムで3年も持てば良いほうだと考えるべきでしょう。

さて，利点・欠点はこんな感じです。これらをどう捕らえるかはプログラマーの自由ですけれど，ボクとしては，システムとしてもプログラムとしても，かなりの魅力を備えているように思えるのです。

構造は非常に簡単！

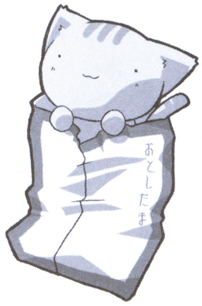
ではでは，つぎは，この言語システムのプログラム的な構造について，簡単にお話をしておきましょう。

言語システムを動作させるには，大きくわけて二つのファイルが必要になります。ひとつは，ゲームの処理を記述するテキスト・ファイル，これをマクロ・ファイルと呼ぶことにします。もうひとつは，マクロ・ファイルを読み込み，その内容を解析しながら実行していく，言語インタプリタです。こちらはBASICやC言語で記述された，普通のプログラムです。

アプリケーションの実行は，言語インタプリタ側を立ちあげることで行います。言語インタプリタは，ウィンドウを作るなど固有の初期化処理を行い，マクロ・ファイルを読み込み，解析処理を開始します。

解析なんて言うと，むずかしいっ，て思っちゃうかもしれませんが，じつはそうでもありません。

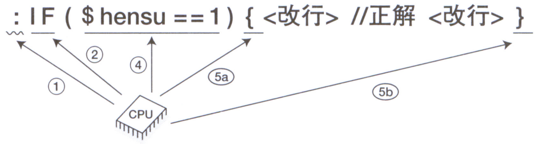
```
:IF( $hensu == 1 ){
        // 正解
}
```



こんなテキストを解析します。
改行が入っていて2次元的に見えますが，
じつはメモリの中では，

```
:IF( $hensu == 1 ){<改行>　// 正解<改行>}
```

というように1次元的に並んでいるのです。これを1byteごとに解析していきます。



①「:」を見つけたCPUは，「ここから命令が始まるんだ」と認識。
②「:」のつぎの文字から英文字が続くところまで見ていくと，「IF」という文字が取得できた。
③「IF」のルーチンへ飛ぶ。IFは，カッコ内に囲われた式を判断する条件判定命令とする。
④カッコ内の式を取得して，それが正しいか判断する。
5a 正しい場合，{ が来るところまでジャンプする。
5b 正しくない場合，} が来るところまでジャンプする。

仕様を決めちゃおう！

構造さえわかれば，さぁいよいよプログラム！　といきたいところですが，まだ大事なことが残っています。そう，マクロの仕様です。この仕様が決まらないと，何を組んでいいのかわからなくなってしまいますね。

ここでは，Windowsだけに限らず，いろいろなプラットフォームで実行することを念頭に入れ，変数や条件判断の処理系を中心に作っていくことにします。名づけて，ベーマガ・マクロ・プロセッサ（仮）。こういうのは雰囲気から入ると，

ベーマガ・マクロ・プロセッサ(仮)
略称「BMP」仕様草案

- 整数値を格納できる変数システム
- 演算処理は四則演算対応
- 条件分岐にIF文。もちろん入れ子対応
- BASICライクにGOTO命令でジャンプ
- サブ・ルーチン処理用にGOSUB/RETURN
- 文字列変数や配列もほしいところ
- デバッグ用に，文字列を画面出力
- マクロ内にコメントを書けるようにする

気合いの入りかたもちがうものです。

たったこれだけ？と思うなかれ。非常に簡素に見えますが，じつはこれだけで，プログラムのおおまかな処理というのが実現できてしまいます。

画面周りや音，入力系などのデバイス関連は，どうしてもそのプラットフォーム独自の仕様を考えざるを得ません。ですから，そのあたりは別命令で用意するとして，プロセッサの中核仕様には含まないものとします。ただ，デバッグ用途として，文字列の画面出力くらいはついていないと，いつまでたっても計算処理を行っているばかりになってしまいますので，これは早めに実装したいものです。

ちなみに，この記事でのサンプル・ソースは，非常に移植性の高いC言語で書いていきます。もちろん，アルゴリズムから紹介していきますから，BASICでの実装もむずかしくないでしょう。

C言語サンプルの理由として，OSに依存しない計算関連の処理は，ほとんどのプラットフォームで，ソースの変更なくコンパイルが可能という点があります。つまり，C言語コンパイラを入手すれば，より多くの環境でベーマガ・マクロ・プロセッサを動かすことが可能なわけ。

文字解析の基本とベース・プログラム

それでは，今回の締めくくりに，文字列解析の基本中の基本である，文字コードについてお話します。

マクロ・ファイルは基本的に，テキスト・エディタやメモ帖で作ることができる，純粋なテキスト・ファイルです。マクロ命令自体は半角英文字で記述することにしますが，パラメータとしての文字列や，コメントには日本語の文字を使いたいものです。

さて，文字というのも，コンピュータ上では数値として表現されています。半角英文字の「A」は41，のように，文字と数値が一対一で対応しているのです。

半角文字については，**第1表**を参照してください。これは，文字コード表と言い，テキスト・ファイルで扱われる半角文字はこの表のとおりになっているのです。

《第1表》ASCII文字コード表

	0	0	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0		SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	HM	CL	CR	SO	SI
1	DE	D1	D2	D3	D4	NK	SN	EB	CN	EM	SB	EC	→	←	↑	↓
2		!	"	#	$	%	&	'	(	)	*	+	,	_	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	_	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8		XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
9	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
a		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
b	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
c	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
d	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
e	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
f	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	

※表は16進数 00 ～ 1f まではコントロールコード XXXX は全角文字の1byte目

問題は，日本語は半角文字だけではなく，全角文字というものが存在すること。第1表を見ればわかるとおり，半角文字の表の中には日本語の漢字やひらがながありません。

しかし，ちゃんとテキスト・ファイルには日本語を書くことができます。これはどういうことかというと，半角文字は1バイトで表現されていますが，全角文字は2バイトで表現され，半角か全角かを判断する方法として，半角文字の1文字目が文字コード表で対応する文字がなければ続く1バイトを加え，2バイトの全角文字として判断する，という手法をとっています。

つまりテキスト文字を解析するときは，1文字目が半角か全角かを判断し，もし全角文字の1バイト目であるなら，続く1バイトもひとつの文字として認識しなくてはいけないのです。

この判定のための範囲は，**第2表**のとおりになっています。

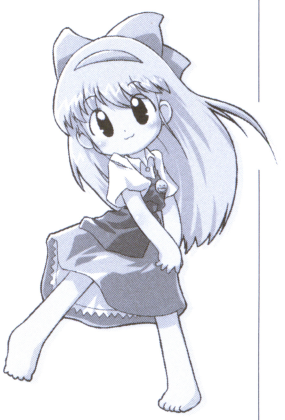
《第2表》半角と全角の割り当て ※数値は16進数

0x00 ～ 0x80	半角
0x81 ～ 0x9f	全角
0xa0 ～ 0xdf	半角
0xe0 ～ 0xfe	全角
0xff	半角

これを踏まえ，テキスト・ファイルを読み込んで1文字ずつ解析していくプログラムは，**リスト1**のようになります。このプログラムが，言語システムのベースになるのです。

《リスト1》言語システムのベースとなるプログラム・リスト

```
#include <stdio.h>
#include <stdlib.h>

// 漢字コード判定用
static int IsKanji( unsigned char cData ) {
        if ( cData < 0x81 ) { return 0; }
        if ( cData < 0xa0 ) { return 1; }
        if ( cData < 0xe0 ) { return 0; }
        if ( cData < 0xff ) { return 1; }
        return 0;
}

void main()
{
        FILE *fp; int nSize, nPnt;
        unsigned char *p;
        fp = fopen( "script.txt", "rb" );
        fseek( fp, 0, SEEK_END ); nSize = ftell( fp ); fseek( fp, 0, SEEK_SET );
        p = ( unsigned char* )malloc( nSize + 1);
        fread( p, 1, nSize, fp );
        fclose( fp );
        nPnt = 0;
        while( nPnt < nSize ) {
                if ( IsKanji( p[nPnt] ) ) {
                        nPnt += 2;
                } else {
                        nPnt++;
                }
        }
        free( p );
}
```



待て！次号

さて来月から，このベース・プログラム(**リスト1**)にどんどん手を加えていって，言語システムを作っていこうと思います。

コマンド追加も柔軟にできるプログラムを考えてますから，ぜひ自分のプログラムに組み込んで，独自言語の世界を楽しんでください。

(無　料)
フリーソフト

Hot Soup Processor HSP完全マスター

楽々3Dプログラミング！

第1回

監修：おにたま

講師:悠黒喧史

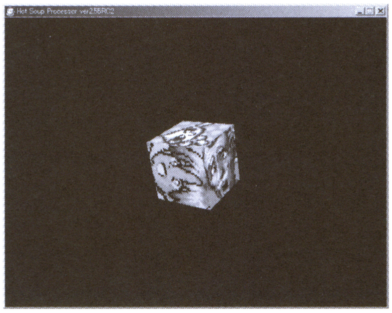
はじめてHSPと出合ったときは，おもしろさのあまりフリーソフトなのにカンパ（千円）。その後，好きが高じてHSPを書籍出版までこぎつけた。HSP Usherを自称する。

「簡単な3Dモデルの表示」

今回のポイント

立方体の表示，移動，回転

HSPはバージョン2.55から3D機能に対応しました。共通の命令でDirectX使用モードとDirectXを使用しないモードを選択できる柔軟なプログラミング環境と，独自の3Dエンジン搭載が特徴となっています。



この講座は，HSPをこれからはじめようというビギナー向けの入門講座というよりも，すでにHSPを使ってプログラミングを楽しんでいて，HSPの新しい機能である3Dプログラミングにトライしたいと思っている中級ユーザーが対象となります。

ただ，これからという人向けに，コーナー最後には，HSPの入力手順と，実行方法を掲載しています。

講座内容は以下のとおり，全6回の予定です。

第1回　簡単な3Dモデルの表示
第2回　MX形式の3Dモデルを作る
第3回　カメラとオブジェクト
第4回　MXモデルでゲーム制作（1）
第5回　MXモデルでゲーム制作（2）
第6回　MXモデルでゲーム制作（3）

3D機能拡張プラグイン[HGIMG]

HSPで3D機能を有効にするには，機能拡張プラグインhgimg.dllを使います。スクリプトの最初に，つぎのようにインクルード命令を記述するだけで，3D命令が使えるようになります。

```
#include "hgimg.as"
```

3Dモデルを表示してみよう

いきなりですが，シンプルな3Dの箱，立方体を画面に表示してみましょう。

3Dモデルを表示するには，まずデザインを用意する処理（モデルの登録）を行い，つぎに，実際に画面に表示するオブジェクトを作成，最後にhgdrawという命令で，画面にオブジェクトの表示処理を行います。

3Dモデル表示の処理手順

［モデル登録］→［オブジェクト作成］→［オブジェクト表示］

立方体モデルの作成には，addbox命令を使います。

《リスト1》立方体表示のサンプル・リスト

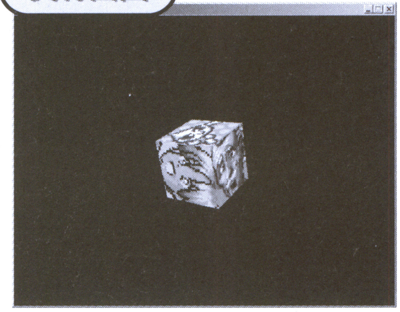
```
#include "hgimg.as"

  hgini                          ; HGIMGの初期化
;
  setsizef 1.2f,1.2f             ; 立方体モデルのXYサイズを指定
  setuv 0,0,31,31                ; テクスチャ画像の1面範囲
  addbox md                      ; 立方体モデルの登録(mdにモデルIDが返る)
  texload "bxtx.bmp"             ; テクスチャーの読込み
;
  regobj ob,md                   ; オブジェクト作成
  selang ob :objset3 2,3,1       ; ちょっと角度をつけて見やすく
;
  gsel 0
  hgdraw                         ; 全オブジェクト描画処理
  hgsync 10                      ; ウェイト
  stop
```

立方体に貼り付けるテクスチャ

bxtx.bmp
（32×96ドット）

※テクスチャ＝3Dモデルの面に貼り付ける画像

実行画面



《リスト2》複数の立方体を表示するプログラム。リスト

```
#include "hgimg.as"

 hgini                                 ; HGIMGの初期化
;―――――――――――――――――――――――――――――――――――――――――――
 setsizef 1.2f,1.2f                    ; 立方体モデルのXYサイズを指定
 setuv 0,0,31,31                       ; テクスチャ画像の1面範囲
 addbox md                             ; 立方体モデルの登録(mdにモデルIDが返る)
 texload "bxtx.bmp"                    ; テクスチャーの読込み
;―――――――――――――――――――――――――――――――――――――――――――
 repeat 3
   regobj ob.cnt,md                    ; オブジェクト作成
   i=ob.cnt
   selang i :objset3 2,3,cnt           ; ちょっと角度をつけて見やすく
   selpos i :objset3 cnt-1*2,0,0       ; 位置を設定
 loop
;―――――――――――――――――――――――――――――――――――――――――――
 gsel 0
 hgdraw                                ; 全オブジェクト描画処理
 hgsync 10                             ; ウェイト
 stop
```

実行画面

リスト1と2の解説

・HGIMGの初期化

最初に必ずhgini命令で，現在のウィンドウIDをもとにHGIMGの初期化を行います。

・モデルの登録

まずsetsizef命令で，これから作る立方体モデルのサイズを指定します。この命令のパラメータは**小数指定**です。小数は0.3fなど末尾にfを付けて記述します。HGIMGで使う命令には小数を使うものが多いので注意してください。

続いてsetuv命令で，これから読み込むテクスチャ画像の立方体1面分のドット範囲を指定します。

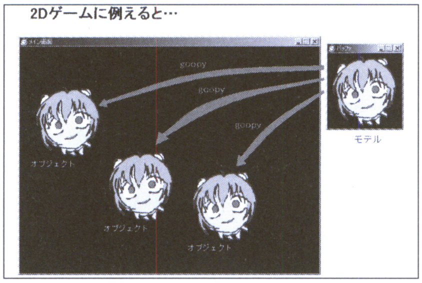
addbox命令で立方体モデルの登録を行い，続くtexload命令で読み込む画像ファイルをテクスチャとして貼り付けます。

以上の手順が，モデルの登録処理となります。

モデルの登録は，単に形状データを用意する処理にすぎません。2Dゲームなどを作るとき，バッファ（ウィンドウID2など）にキャラクター画像をあらかじめ読み込んでおく方法に似ています。

登録したモデルには，登録順にモデルID番号が自動的に割りふられます。

・モデルとオブジェクトの関係



使用した命令

setsizef	モデルサイズの指定
setuv	テクスチャの画像範囲を指定
addbox	立方体モデルの登録
texload	テクスチャとする画像ファイルの読み込み

・オブジェクトの作成

regobj命令で，指定したモデルIDの形状をしたオブジェクトを作成することができます。

一つのモデルが登録されていれば，オブジェクトはいくつでも作成することができ，作成したオブジェクトには順次，オブジェクトID番号が自動で割りふられます。

オブジェクトにはMOC（モーション・オブジェクト・コントローラ）情報という変数のグループが設けられており，この変数に足したり引いたりの計算を加えることで，オブジェクトの角度や座標を自由に変更することができるようになっています。たとえば，オブジェクトの座標を変更する場合には，selpos命令で，指定したオブジェクトIDの座標値にアクセスすることができます。

命令	MOC情報
selpos	X,Y,Z座標値をアクセス対象にする
selang	X,Y,Z角度値をアクセス対象にする
selscale	X,Y,Z拡大率をアクセス対象にする
seldir	X,Y,Z移動量をアクセス対象にする

・スクリプト記述例

selpos ob :objset3 0,0,0

objset3命令は，現在，対象となっているMOC情報に整数値をセットします。

この例では，変数obの値を指定IDとするオブジェクトの座標を（0,0,0）にセットしています。

このようにobj系の命令は，sel系の命令と組み合わせて使います。objaddやobjmovなど，MOCの値を変更するための命令が多数あります。

使用した命令

regobj	オブジェクトを作成
selang	指定されたオブジェクトのMOC角度を対象にする
selpos	指定されたオブジェクトのMOC座標を対象にする
objset3	MOC情報に整数値をセット

・オブジェクトの表示

regobj命令で作成されたすべてのオブジェクトは，hgdraw命令ひとつで画面にすべて描画されます。

そして，hgdraw命令のあとには必ずhgsync命令でウェイト処理を入れてください。ウェイトがかからないと描画された画面の更新が行われません。これはWindowsにタスクを渡すための処理で，hgsync命令は，HGIMGでの処理においてawait命令にあたるものです。

使用した命令

hgdraw	すべてのオブジェクトを描画
hgsync	ウェイト処理を行う

・立方体を移動させてみよう

では，この立方体を上下左右前後に移動できるようにしてみましょう。

オブジェクトIDを指定し，そのMOC情報の中にある座標値を変更すれば，オブジェクトを移動させることができます。

《リスト3》立方体の移動プログラム・リスト

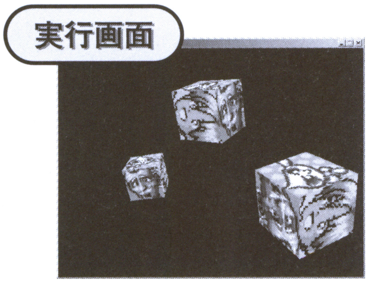
```
#include "hgimg.as"

 hgini                             ; HGIMGの初期化
;
 setsizef 1.2f,1.2f                ; 立方体モデルのXYサイズを指定
 setuv 0,0,31,31                   ; テクスチャ画像の1面範囲
 addbox md                         ; 立方体モデルの登録(mdにモデルIDが返る)
 texload "bxtx.bmp"                ; テクスチャーの読込み
;
 regobj ob,md                      ; オブジェクト作成
 selang ob :objset3 2,3,1          ; ちょっと角度をつけて見やすく
;
 gsel 0
*main
 selpos ob                         ; MOC座標にアクセス

 stick ky,79                       ; キースキャン

 if ky&1 :objaddf1 0,-0.1f         ; X軸移動
 if ky&4 :objaddf1 0,0.1f          ;
 if ky=2 :objaddf1 1,-0.1f         ; Y軸移動
 if ky=8 :objaddf1 1,0.1f          ;
 if ky=66 :objaddf1 2,-0.1f        ; Z軸移動
 if ky=72 :objaddf1 2,0.1f         ;

 hgdraw                            ; 全オブジェクト描画処理
 hgsync 10                         ; ウエイト

 if ky!128 :goto *main             ; [Esc]が押されるまでループ

 hgbye                             ; HGIMGを解放
```

カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

[Esc]でプログラム終了。

《リスト4》複数の立方体の移動プログラム・リスト

```
#include "hgimg.as"

 hgini                             ; HGIMGの初期化
;
 setsizef 1.2f,1.2f                ; 立方体モデルのXYサイズを指定
 setuv 0,0,31,31                   ; テクスチャ画像の1面範囲
 addbox md                         ; 立方体モデルの登録(mdにモデルIDが返る)
 texload "bxtx.bmp"                ; テクスチャーの読込み
;
 repeat 3
   regobj ob.cnt,md                ; オブジェクト作成
   i=ob.cnt
   selang i :objset3 2,3,cnt       ; ちょっと角度をつけて見やすく
   selpos i :objset3 cnt-1*2,0,0   ; 位置を設定
 loop
;
 gsel 0 :i=1
*main
 selpos ob.i                       ; MOC座標にアクセス

 stick ky,79 :if ky=1024 :i=i+1\3  ; オブジェクトIDの切換え

 if ky&1 :objaddf1 0,-0.1f         ; X軸移動
 if ky&4 :objaddf1 0,0.1f          ;
 if ky=2 :objaddf1 1,-0.1f         ; Y軸移動
 if ky=8 :objaddf1 1,0.1f          ;
 if ky=66 :objaddf1 2,-0.1f        ; Z軸移動
 if ky=72 :objaddf1 2,0.1f         ;

 hgdraw                            ; 全オブジェクト描画処理

 color 255,255,255                 ; オブジェクトIDの表示
 pos 8,8 :mes "ObjectID="+ob.i     ;

 hgsync 10                         ; ウエイト

 if ky!128 :goto *main             ; [Esc]が押されるまでループ

 hgbye                             ; HGIMGを解放
```



カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

[Tab]で対象となるオブジェクトIDを切り換えます。

[Esc]でプログラム終了。

使用した命令

objaddf1　MOC情報のX,Y,Zのいずれか一つを指定し，小数値をセット

hgbye　HGIMGの解放（HGIMGの終了処理）

・立方体を回転させてみよう

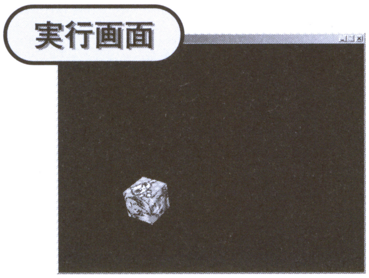
さらに，この立方体を回転させる処理を加えてみましょう。

回転させるオブジェクトIDを指定し，そのMOC情報の中にある角度値を変更します。角度値はラジアンで指定します。

つまり，3.14で180度（半回転）になります。

《リスト5》立方体の回転プログラム・リスト

```
#include "hgimg.as"

 hgini                             ; HGIMGの初期化
;
 setsizef 1.2f,1.2f                ; 立方体モデルのXYサイズを指定
 setuv 0,0,31,31                   ; テクスチャ画像の1面範囲
 addbox md                         ; 立方体モデルの登録(mdにモデルIDが返る)
 texload "bxtx.bmp"                ; テクスチャーの読込み
;
 regobj ob,md                      ; オブジェクト作成
 selang ob :objset3 2,3,1          ; ちょっと角度をつけて見やすく
;
 gsel 0
*main
 selpos ob                         ; MOC座標にアクセス

 stick ky,79                       ; キースキャン

 if ky&1 :objaddf1 0,-0.1f         ; X軸移動
 if ky&4 :objaddf1 0,0.1f          ;
 if ky=2 :objaddf1 1,-0.1f         ; Y軸移動
 if ky=8 :objaddf1 1,0.1f          ;
 if ky=66 :objaddf1 2,-0.1f        ; Z軸移動
 if ky=72 :objaddf1 2,0.1f         ;

 selang ob                         ; MOC角度にアクセス

 k=-1                              ; テンキー入力チェック
 repeat 10                         ;
   getkey a,96+cnt                 ;
   if a=1 :k=cnt :break            ;
 loop                              ;

 if k=8 :objaddf1 0,-0.1f          ; X軸回転
 if k=2 :objaddf1 0,0.1f           ;
 if k=4 :objaddf1 1,-0.1f          ; Y軸回転
 if k=6 :objaddf1 1,0.1f           ;
 if k=1 :objaddf1 2,-0.1f          ; Z軸回転
 if k=0 :objaddf1 2,0.1f           ;

 hgdraw                            ; 全オブジェクト描画処理
 hgsync 10                         ; ウエイト

 if ky!128 :goto *main             ; [Esc]が押されるまでループ

 hgbye                             ; HGIMGを解放
```



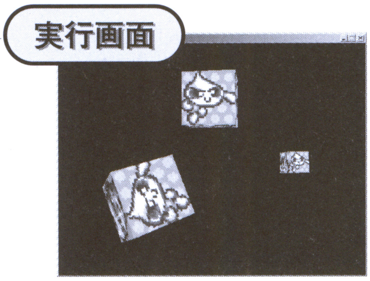
カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

テンキーの[4][6]でY軸回転，[8][2]でX軸回転，[1][0]でZ軸回転をします。

[Esc] でプログラム終了。

《リスト6》複数の立方体の回転プログラム・リスト

```
#include "hgimg.as"

 hgini                             ; HGIMGの初期化
; ----------------------
 setsizef 1.2f,1.2f                ; 立方体モデルのXYサイズを指定
 setuv 0,0,31,31                   ; テクスチャ画像の1面範囲
 addbox md                         ; 立方体モデルの登録(mdにモデルIDが
返る)
 texload "bxtx.bmp"                ; テクスチャーの読込み
; ----------------------
 repeat 3
   regobj ob.cnt,md                ; オブジェクト作成
   i=ob.cnt
   selang i :objset3 2,3,cnt       ; ちょっと角度をつけて見やすく
   selpos i :objset3 cnt-1*2,0,0   ; 位置を設定
 loop
; ----------------------
 gsel 0 :i=1
*main
 selpos ob.i                       ; MOC座標にアクセス

 stick ky,79 :if ky=1024 :i=i+1\3  ; オブジェクトIDの切換え

 if ky&1 :objaddf1 0,-0.1f         ; X軸移動
 if ky&4 :objaddf1 0,0.1f          ;
 if ky=2 :objaddf1 1,-0.1f         ; Y軸移動
 if ky=8 :objaddf1 1,0.1f          ;
 if ky=66 :objaddf1 2,-0.1f        ; Z軸移動
 if ky=72 :objaddf1 2,0.1f         ;

 selang ob.i                       ; MOC角度にアクセス

 k=-1                              ; テンキー入力チェック
 repeat 10                         ;
   getkey a,96+cnt                 ;
   if a=1 :k=cnt :break            ;
 loop                              ;

 if k=8 :objaddf1 0,-0.1f          ; X軸回転
 if k=2 :objaddf1 0,0.1f           ;
 if k=4 :objaddf1 1,-0.1f          ; Y軸回転
 if k=6 :objaddf1 1,0.1f           ;
 if k=1 :objaddf1 2,-0.1f          ; Z軸回転
 if k=0 :objaddf1 2,0.1f           ;

 hgdraw                            ; 全オブジェクト描画処理

 color 255,255,255                 ; オブジェクトIDの表示
 pos 8,8 :mes "ObjectID="+ob.i     ;

 hgsync 10                         ; ウエイト

 if ky!128 :goto *main             ; [Esc]が押されるまでループ

 hgbye                             ; HGIMGを解放
```

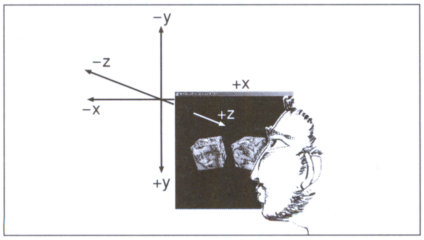


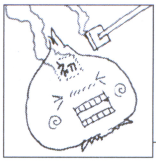
カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

テンキーの[4][6]でY軸回転，[8][2]でX軸回転，[1][0]でZ軸回転をします。

[Tab]で対象となるオブジェクトIDを切り換えます。[Esc]でプログラム終了。

・HGIMGの3D空間図





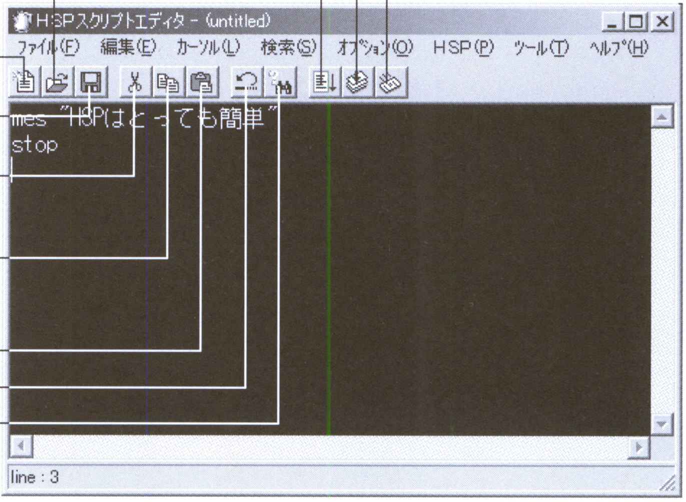
はじめて使う人のための，HSPでプログラムを入力する方法

HSPでは，プログラムを記述する命令文をスクリプトと呼びます。

HSPをインストールしたフォルダにある**hsed2.exe**をダブルクリックすると，スクリプトを入力するための専用エディター「HSPスクリプトエディター」が開きます。

デスクトップに**hsed2.exe**へのショートカットを作っておくと便利です。

HSPスクリプトエディター



外部ファイル実行ボタン
HSPスクリプトエディター以外の，たとえば，使い慣れたエディタなどで編集するスクリプトを実行することができます

オブジェクト作成ボタン
実行形式のファイルを作る際にオブジェクト・ファイルを作成します

HSP実行ボタン
編集中のスクリプトを実行します

ロードボタン
スクリプトファイルを読み込みます

新規ボタン
スクリプトを新規入力します

セーブボタン
スクリプトファイルを保存します

カットボタン
選択範囲を切り取ってクリップボードにコピーします

コピーボタン
選択範囲をクリップボードにコピーします

ペーストボタン
クリップボードの内容を貼り付けます

アンドゥボタン
直前の状態に戻します。編集ミスなどしたときに使います

検索ボタン
編集中のスクリプトから，指定した文字列を検索します

プログラムの実行

実行画面

HSP実行ボタンをクリックすると，その場でプログラムを実行できます。

ウインドウの［×］をクリックすると，プログラムは終了します。

基本的にプログラムというのは，書かれた命令文を上から順番に実行していきます。

mesは，指定された文字を画面に表示する命令。**stop**はプログラムをそこで停止させる命令です。

※HSPの命令文＝ここでは二つの命令しか使っていませんが，HSPには，非常に多くの命令が用意されています。HSPスクリプトエディターのヘルプメニューから**命令リファレンスを開く**を選ぶと，命令の一覧と，その説明を見ることができます。一つ一つの命令を試しながら憶えていくといいでしょう。

スクリプトの入力

文字入力位置にキャレットが点滅しているはずです。カンタンなスクリプトプログラムを入力してみましょう。命令文はすべて半角英数字で入力します。

※キャレット＝文字入力位置を示すポインタ

《リスト1》サンプル・プログラム

```
mes "HSPはとっても簡単"
stop
```

スカッシュゲームのスクリプトにチャレンジ！

スクリプトの入力と実行の手順に少し慣れてきたら，つぎのリストにもチャレンジしてみてください。

カーソル・キーの左右でラケットを動かして，ボールを受け返す，簡単なスカッシュゲームのプログラムです。

《リスト2》スカッシュゲームのプログラム・リスト

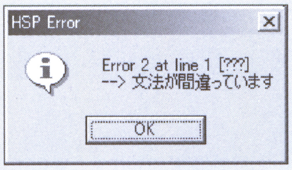
```
 screen 0,320,320                                  ; スクリーンサイズ
 title "スカッシュゲーム"                          ; タイトルバーの表示
 font "FixedSys",16                                ; フォント
 px=160                                            ; ラケットのX座標
 x=160 :y=160                                      ; ボールのX,Y座標
 dx=8 :dy=-8                                       ; ボールの移動値
*main
 stick ky,5                                        ; キースキャン
 if (ky=1)&(px>0) :px-=8                           ; ラケットを左に移動
 if (ky=4)&(px<320) :px+=8                         ; ラケットを右に移動
 x+=dx :y+=dy                                      ; ボールの移動
 if (x<0)|(x>320) :dx=-dx                          ; 左右の壁で跳ね返る
 if y<0 :dy=-dy                                    ; 上の壁で跳ね返る
 if y>320 :pos 104,152 :mes "ゲームオーバー" :stop  ; 下に行ったらゲーム終了
 if (y-296/8=0)&(x-px/32=0) :dy=-dy                ; ラケットなら跳ね返る
 redraw 0                                          ; 画面を非更新モードに
 color 0,0,0 :boxf                                 ; 画面を黒でクリア
 color 255,255,255                                 ; カラー設定を白に
 pos px-24,296 :mes "■■■"                          ; ラケット表示
 pos x-8,y-8 :mes "●"                              ; ボール表示
 redraw 1                                          ; 画面を更新モードに
 await 50                                          ; 0.5秒ウェイト
 goto *main                                        ; ラベル「*main」へジャンプ
```

スクリプトのエラー

HSP実行ボタンをクリックすると一瞬でコンパイルして実行しますが，スクリプトに何らかのまちがいがあると，エラーを表示して中断します。

例）

```
mos "HSPはとっても簡単"
stop
```



エラー表示

この例では，1行目の"mos"がスペルミスでまちがっています。このような命令文は存在しないのでエラーになります。

※コンパイル＝スクリプトをコンピュータが直接実行できる形式に変換する処理

さらに欲張りな人のための，HSPで実行ファイルの作成方法

HSPは，HSPがインストールされていないパソコンでも単独で実行できるEXEファイルを作成することができます。

オブジェクト・ファイルの作成

EXEにしたいスクリプトをHSPスクリプトエディターに読み込み，HSPメニューからSTART.AXファイル作成を選択し，オブジェクト・ファイルを作成します。

オブジェクト・ファイルには，拡張子が"ax"と付きます。

必要なファイルをパックする

プログラムで使用しているファイルを一つにパックします。

ツールメニューからPACKFILE編集を選択し，PACKFILE編集のダイアログを開きます。

・PACKFILE編集のダイアログ

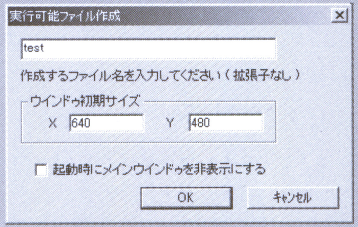
start.ax　スクリプトから作成したオブジェクト・ファイル

パックするファイルを追加ボタンで選び出し，閉じるボタンをクリックしてダイアログを閉じます。

EXEファイルの作成

ツールメニューからEXEファイル作成を選択し，実行可能ファイル作成のダイアログを開きます。

・実行可能ファイル作成のダイアログ



作成するEXEファイルに名前を付け，OKボタンをクリックすると作成されます。

・実行ファイルの完成！

HGIMG-FAQ

質　問

立方体の他にどんな3Dモデルが表示できますか？

答　え

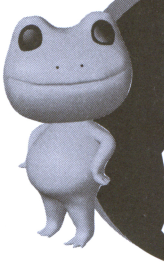
プレート（板）の表示も簡単にできます。プレートとは平面の板のこと。プレートを表示するための処理手順は，立方体での説明と同じです。addbox命令に代わってaddplate命令を使います。リスト4などを変更して試してみるとよいでしょう。

ちなみに2次元の平面板なので，指定する角度によっては見えづらくなることがあります。

複雑な形状のポリゴンモデルも扱えますが，これについては，次回くわしく解説します。

お楽しみに！

プログラマーをめざす人の必見講座



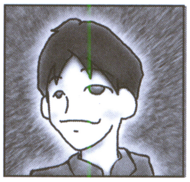
図解！Visual Basic ゲーム作成術

■敵とプレイヤーの力関係を逆転させるテクニック！

第11回　キャラクター・アクション・ゲーム④プレイヤー・キャラクターの逆襲

キャラクター・アクション・ゲームでは，たいていの場合「プレイヤー・キャラクターは敵キャラクターに追いかけられたり攻撃される」ということが普通です。しかし，条件に応じてプレイヤー・キャラクターが反撃できるようにしているゲームも多く見られます。反撃できる条件をうまく工夫することで，反撃できたときの爽快感を高めたり，ゲームを奥深いものにしたりできます。今回は，このような逆襲要素を『Microsoft Visual Basic（以下VBと略）』で作成してみましょう。

TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。まさに流星“雨”だった，今回の獅子座流星群。1週間前から朝型の生活になじませ，ウォッチング・ポイントまで出向いていったカイがあった。大量のお願いごとが，かなってくれればいいのだけど…。

Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）

オープン価格

※編集部の価格調査によると，下記価格が目安になります。

ラーニング版2万円強

（アカデミック・パッケージ1万円弱）

プロフェッショナル版8～9万円

（アカデミック・パッケージ1万円強）

問い合わせ先

マイクロソフト

☎03-5454-2300

ホームページ

http://www.microsoft.com/japan/

動作環境

Windows 95／98／2000／NT4.0が動作するマシン

パックマンで確立されたプレイヤーの逆襲！

多くのキャラクター・アクション・ゲームでは，プレイヤー・キャラクターは敵キャラクターよりも弱い存在で，敵キャラクターに追跡されたり攻撃されたりするようになっています。しかし，追いかけられっぱなしや攻撃されっぱなしというのは，気分的にはあまりよいものとはいえませんね。プレイヤーが，ゲームにストレスを感じてしまう場合もあります。

これに対応するために，プレイヤー・キャラクターが敵キャラクターに反撃できるタイミングや条件を用意し，プレイヤー・キャラクターと敵キャラクターの立場が反転するようにしたゲームもいくつか存在します。

代表例として，『パックマン』があります。通常モンスターよりも弱い立場にあるパックマンが，パワーアップ・エサを食べることで一定期間はモンスターよりも強い立場となり，モンスターを食べられるようになっています。このような「立場の反転状態」を付け加えることで，プレイヤーに対しては反撃時に爽快感をあたえられます。

さらに，パックマンの場合は反撃時にモンスターを食べると高得点となり，モンスターを引きつけてからパワーアップするのが有効になっています。これは，高得点を狙うためにはモンスターを引きつけなければならず，一歩まちがえればミスにつながるという諸刃の剣になっています。このような「ハイリスク＆ハイ・リターン」な要素を程度を調整してゲームに組み込めば，ゲーム性を高められるのです。

“モード”でキャラクター関係を変更する！

プレイヤー・キャラクターに反撃させるためには，キャラクターの相互関係を変更しなければなりません。

通常は，プレイヤー・キャラクターと敵キャラクターが接触した場合，プレイヤー・キャラクターがダメージを受けたりミスになったりするように処理しますね。しかし，反撃中は敵キャラクターがダメージを受けたり消滅したりするように処理しなければなりません。

このようなプログラムの処理の状態のことを“モード”と呼び，処理内容が変わる場合は「モードが変わる」という表現をします。ここでは，

●通常モード：プレイヤー・キャラクターが弱い立場の通常状態

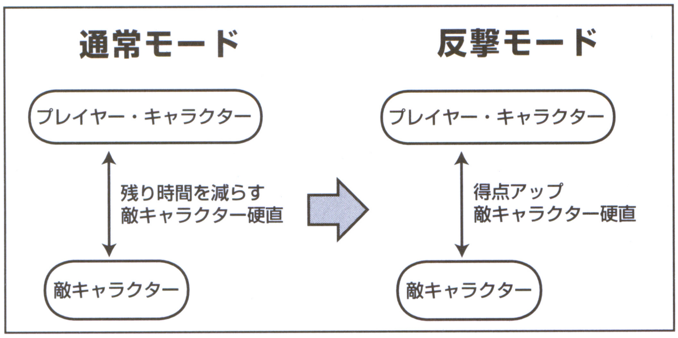
●反撃モード：プレイヤー・キャラクターが反撃できる状態

と表現しましょう。

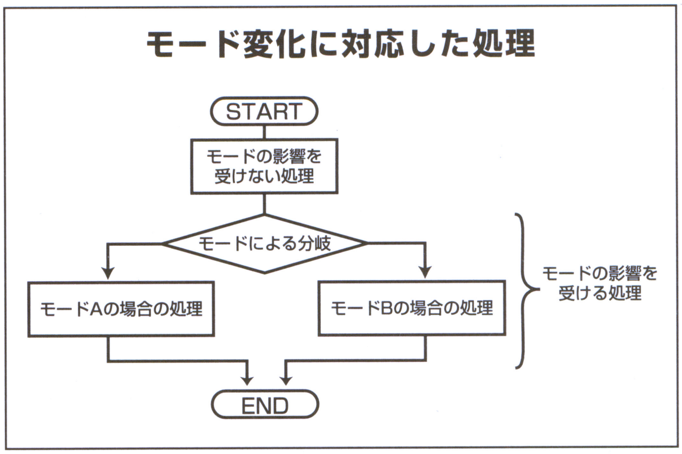
通常モードでは，プレイヤー・キャラクターよりも敵キャラクターが強い関係にあります。接触した場合は，プレイヤー・キャラクターにダメージをあたえ，敵キャラクターはしばらく硬直させるという処理を行います。

これに対し，反撃モードではキャラクターの関係が**第1図**のように変化し，プレイヤー・キャラクターが敵キャラクターよりも強い関係になります。接触した場合は，敵キャラクターを硬直させ，得点をアップさせる処理を行います。硬直した敵キャラクターは反撃モード終了後，復活させます。

このようなモード変化をプログラミン



《第1図》モードにより，キャラクター関係を変化させる



《第2図》モードの影響を受けない処理はつねに実行し，影響を受ける場合は分岐処理する

グする場合は，モード状態を示す変数modecを用意し，この変数の値に応じて処理を分岐させます。

キャラクターの処理には，モードの影響を受けない処理と受ける処理があります。影響を受けない処理については常時実行するようにし，影響を受ける部分について処理を分岐させてやります（**第2図**）。

このようにすれば，モードの追加に対応する部分だけを用意することで，モード変化に対応した処理プログラムができあがります。

プレイヤー・キャラクターから逃げるには？

敵キャラクターは，通常モード時にはプレイヤー・キャラクターを追跡する動作をします。

しかし，反撃モードのときにも同じような動きをしていると，プレイヤーが敵キャラクターを追いかけなくても，敵キャラクターのほうからプレイヤー・キャラクターに体当たりしてきてくれることになります。これではゲームとして問題があるので，反撃モードのときには，敵キャラクターにプレイヤー・キャラクターから逃げるような動きをさせたいところです。

特定のキャラクターから逃げる動きというのは，特定のキャラクターを追跡する動きの正反対の動きになります。このため，追跡プログラムが用意できていれば，逃避プログラムも簡単に作れます。

追跡プログラムでは，敵キャラクターから見たプレイヤー・キャラクターの方向をもとに，進むべき方向をきめていました。プレイヤー・キャラクターが右側に見えているならば，右のほうに曲がるような動きをさせるわけです。

これと反対に，プレイヤー・キャラクターが右側に見えているときに左方向に曲がるようにすれば，プレイヤー・キャラクターから離れる動きになります（**第3図**）。これが，そのまま逃避プログラムとなるのです。

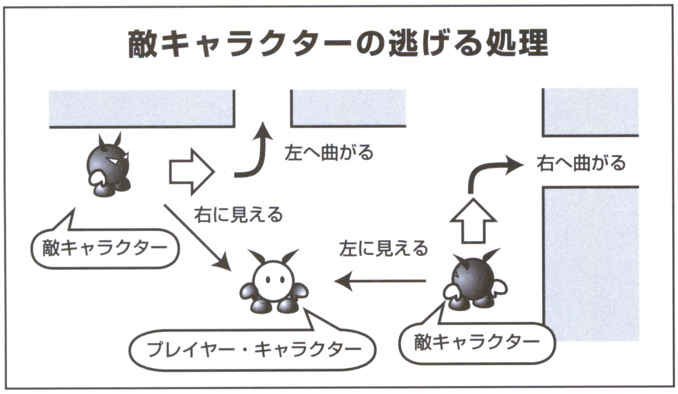
テーブルの値をモードで変更する！

先月までの追跡プログラムでは，移動方向探索順を示すテーブルehtを使っていました。このehtを拡張し，通常モードと反撃モードでは異なるテーブル値になるようにしてやります。

ehtは2次元配列変数で，プレイヤー・キャラクターの方向と，優先順位が添え字になっています。これに，モードを指定する添え字を追加し，3次元配列変数に拡張します。

もともと2次元配列になっている構成のehtを，**第4図**のようにモードによってもデータが変わるようにしてやるわけです。通常モードの場合のデータは従来どおりとし，反撃モードの場合のデータを逃避するようなデータにしてやります。これで，追跡プログラムはほとんどそのままの形で，モードに応じて追いかけたり逃げたりするプログラムになるわけです。

テーブルを活用していない場合は，このようなモードの追加を行うとプログラム自体が追加されることが多いでしょう。しかし，テーブルを活用していれば，テ



《第3図》プレイヤー・キャラクターが見える方向と逆の方向へ曲がると，逃げる動きになる

モードによるテーブル値の変化

プレイヤー・キャラクターの見える方向

	最優先	2	3	4
正面	前	右	左	後ろ
右	右	前	左	後ろ
後ろ	前	左	右	後ろ
左	左	前	右	後ろ

追跡パターン

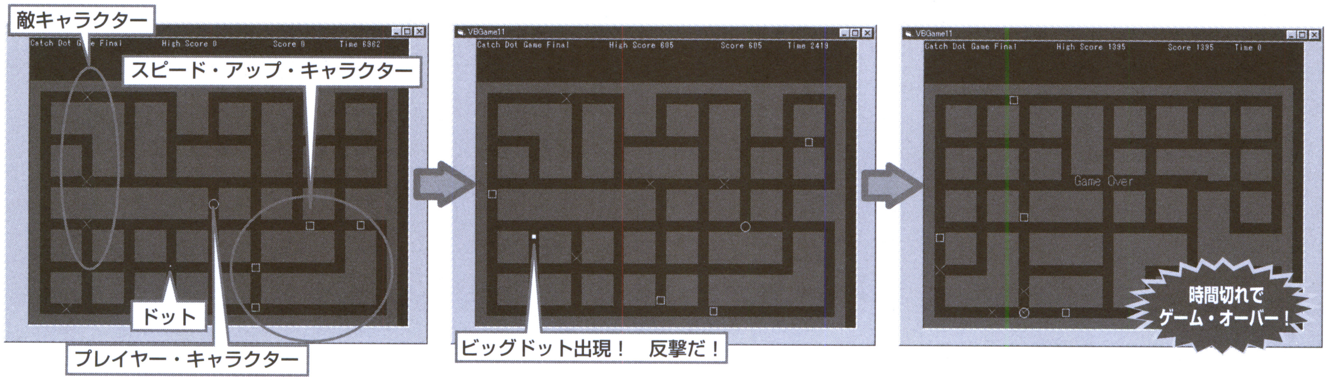
モードによる切替

	最優先	2	3	4
正面	前	左	右	後ろ
右	左	前	右	後ろ
後ろ	前	右	左	後ろ
左	右	前	左	後ろ

逃避パターン

《第4図》変数ehtを拡張して，逃避パターンを追加する

これが『キャッチ・ドットゲームFinal』だ！



《第5図》『キャッチ・ドットゲームFinal』。ビッグドットによる反撃は，条件によっては高得点を狙えるが，敵キャラクターの逃げ足は速いため全敵キャラクターを捕まえることはむずかしい。ドットの得点は残り時間にほぼ比例していることから，ゲーム前半は比較的高い得点になる。よって，前半はドット重視でビッグドットを温存しておき，ドットの得点が落ちてくるゲーム後半で反撃に移るのが効果的だ

ーブル周辺の簡単な変更で機能を追加できる場合もあるわけです。覚えておきましょう。

これがキャラクター・アクションの最後のゲームだ！

それでは，実際にVBでプログラミングされたキャラクター・アクション・ゲーム『キャッチ・ドットゲームFinal（ファイナル）』を実行してみましょう。プログラムは，リスト1です。

リストを入力して実行すると，**第5図**のような画面が表示されます。画面には，プレイヤーが操作するプレイヤー・キャラクターと，敵キャラクターが四つ登場します。敵キャラクターは，ランダムに格子状の通路を移動します。移動した後に，ドットを落としていく場合があります。

プレイヤーは，プレイヤー・キャラクターをテンキーの[2][4][6][8]で上下左右に移動させ，敵キャラクターが落としたドットを制限時間内になるべくたくさん拾い集めてください。ドットを取ると，その段階での残り時間に応じた得点が加算されます。残り時間が多いほど，高得点になるわけです。

ただし，敵キャラクターと接触すると，残り時間が大幅に減ってしまいます。接触された敵キャラクターは，しばらくの間停止します。

ゲーム・フィールド内には，プレイヤー・キャラクターを一定時間の間高速化するスピード・アップ・キャラクターが移動しています。プレイヤー・キャラクターがこのキャラクターに触れると，それ以後一定期間，移動速度が高速化されます。アルゴリズムの拡張によって，敵キャラクターの追跡が厳しくなっているため，スピード・アップ・キャラクターをうまく活用する必要があります。

敵キャラクターは，まれに大きなドット（ビッグドット）を落としていきます。このドットを取ると反撃モードになり，プレイヤー・キャラクターと敵キャラクターの関係が逆転します。反撃モードの間に，敵キャラクターと接触すると敵キャラクターが消滅し，得点が加算されます。一定時間が経過すると，反撃モードは終了します。

リストのここに注目！

リスト1では，通常モードと反撃モードの区別を変数modecで管理しています。

modecは通常0ですが，ビッグドットを取ることで一定数値に設定されます。その後，1ずつ減少していき，再び0にもどります。カウント・ダウンしている間の0でない状態が，反撃モードを示すことになります。

敵キャラクターの移動方向決定部分では，modecが0か0でないかを判断し，変数mを設定します。その変数mを，配列変数ehtの三つ目の添え字として使います。

配列変数ehtは，三つ目の添え字が0のときには追跡するような移動方向を，1のときには逃避するような移動方向をそれぞれ格納しています。変数mの設定を変えることで，プログラムをほとんど変えることなく追跡動作と逃避動作の切り替えが可能になるのです。

移動方向決定処理が単純なテーブル参

《第1表》『キャッチ・ドットゲームFinal』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	プレイヤー・キャラクターX座標
py	プレイヤー・キャラクターY座標
pm	プレイヤー・キャラクター移動カウンタ初期値
pmc	プレイヤー・キャラクター移動カウンタ
spupc	スピード・アップ期間カウンタ
modec	パワーアップ期間カウンタ
ef ()	敵キャラクター状態フラグ
ex ()	敵キャラクターX座標
ey ()	敵キャラクターY座標
eh ()	敵キャラクターの方向
em ()	敵キャラクター移動カウンタ初期値
emc ()	敵キャラクター移動カウンタ
sf ()	スピード・アップ・キャラクター状態フラグ
sx ()	スピード・アップ・キャラクターX座標
sy ()	スピード・アップ・キャラクターY座標
sh ()	スピード・アップ・キャラクターの方向
sm ()	スピード・アップ・キャラクター移動カウンタ初期値
smc ()	スピード・アップ・キャラクター移動カウンタ
dx ()	方向→X座標移動量変換テーブル
dy ()	方向→Y座標移動量変換テーブル
eht ()	敵キャラクター＆スピード・アップ・キャラクター移動方向検索順テーブル
dp ()	キャラクター位置関係テーブル
kg1 ()	敵キャラクター用仮想画面（0：存在しない，それ以外：存在するキャラクターの添え字＋1）
kg2 ()	スピード・アップ・キャラクター用仮想画面（0：存在しない，それ以外：存在するキャラクターの添え字＋1）
kg3 ()	静止キャラクター用仮想画面（0：空白，1：壁，2：ドット，3：ビッグドット）
kg4 ()	敵キャラクター追跡用仮想画面
ti	残り時間
dotc	ドットを発生させる間隔計測カウンタ
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

照になっているため，このような機能拡張ができるようになっています。

まとめ

テーブル処理は，テーブル全体を記憶しておく必要があることから，メモリを大量に消費する傾向があります。10年ほど前はメモリ容量にも制限があったため，無制限にテーブル処理を導入できませんでした。

しかし，今は1Gバイトのメモリも珍しくなく，Windows 2000やWindows XPを使えばそのメモリをフルに活用できます。少々のムダは問題になりませんから，テーブル処理を活用しやすくなっています。

みなさんも，独自のキャラクター・アクション・ゲーム作成に挑戦してみてくださいね。

それでは　また来月！

★本ページのプログラムはhttp://www.basicmagazine.netからダウンロードできます。

《リスト1》『キャッチ・ドットゲームFinal』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用キーの状態を記
録する変数
Dim px%, py%, ph%, pf%, pmc%, pm%, pm2%, spupc% 'プレイヤーキャラクタ
Dim ex%(3), ey%(3), eh%(3), ef%(3), emc%(3), em%(3) '敵キャラクタ
Dim sx%(3), sy%(3), sh%(3), sf%(3), smc%(3), sm%(3) 'スピードアップキャラ
クタ
Dim kg1(35, 26) '仮想画面１（敵キャラクタ用）
Dim kg2(35, 26) '仮想画面２（スピードアップキャラクタ用）
Dim kg3(35, 26) '仮想画面３（ドット＆壁用）
Dim kg4(3, 35, 26) '仮想画面４（敵キャラクタ追跡用）
Dim eht(3, 3, 2) '敵キャラクタ移動方向検索テーブル
Dim dx(3), dy(3) '方向→移動量変換テーブル
Dim dp(3, 2, 2) 'キャラクタ位置関係テーブル
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim dotc% 'ドットを発生させる間隔を数えるカウンタ
Dim modec% 'パワーアップ期間を数えるカウンタ
Dim gamestat% 'ゲーム進行状況

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, xx%, yy%, i%, j%, f%, h%, c0%, c1%
    With Form1
        .Width = Screen.TwipsPerPixelX * 640 'フォームのサイズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame11"
    End With
    With Picture1
         .Width = Screen.TwipsPerPixelX * 576 'ピクチャボックスのサイズを
設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    dx(0) = 0: dy(0) = -1 '移動方向→移動量のテーブル
    dx(1) = 1: dy(1) = 0
    dx(2) = 0: dy(2) = 1
    dx(3) = -1: dy(3) = 0

    dp(0, 0, 0) = 3: dp(0, 0, 1) = 3: dp(0, 0, 2) = 3 '位置関係テーブル
    dp(0, 1, 0) = 0: dp(0, 1, 1) = 0: dp(0, 1, 2) = 2
    dp(0, 2, 0) = 1: dp(0, 2, 1) = 1: dp(0, 2, 2) = 1

    dp(1, 0, 0) = 3: dp(1, 0, 1) = 0: dp(1, 0, 2) = 1
    dp(1, 1, 0) = 3: dp(1, 1, 1) = 0: dp(1, 1, 2) = 1
    dp(1, 2, 0) = 3: dp(1, 2, 1) = 2: dp(1, 2, 2) = 1

    dp(2, 0, 0) = 1: dp(2, 0, 1) = 1: dp(2, 0, 2) = 1
    dp(2, 1, 0) = 2: dp(2, 1, 1) = 0: dp(2, 1, 2) = 0
    dp(2, 2, 0) = 3: dp(2, 2, 1) = 3: dp(2, 2, 2) = 3

    dp(3, 0, 0) = 1: dp(3, 0, 1) = 0: dp(3, 0, 2) = 3
    dp(3, 1, 0) = 1: dp(3, 1, 1) = 0: dp(3, 1, 2) = 3
    dp(3, 2, 0) = 1: dp(3, 2, 1) = 2: dp(3, 2, 2) = 3

    '敵キャラクタ移動方向検索順テーブル
    eht(0, 0, 0) = 0: eht(0, 1, 0) = 1: eht(0, 2, 0) = 3: eht(0, 3, 0) =
2
    eht(1, 0, 0) = 1: eht(1, 1, 0) = 0: eht(1, 2, 0) = 3: eht(1, 3, 0) =
2
    eht(2, 0, 0) = 0: eht(2, 1, 0) = 3: eht(2, 2, 0) = 1: eht(2, 3, 0) =
2
    eht(3, 0, 0) = 3: eht(3, 1, 0) = 0: eht(3, 2, 0) = 1: eht(3, 3, 0) =
2
    eht(0, 0, 1) = 0: eht(0, 1, 1) = 3: eht(0, 2, 1) = 1: eht(0, 3, 1) =
2
    eht(1, 0, 1) = 3: eht(1, 1, 1) = 0: eht(1, 2, 1) = 1: eht(1, 3, 1) =
2
    eht(2, 0, 1) = 0: eht(2, 1, 1) = 1: eht(2, 2, 1) = 3: eht(2, 3, 1) =
2
    eht(3, 0, 1) = 1: eht(3, 1, 1) = 0: eht(3, 2, 1) = 3: eht(3, 3, 1) =
2

    Picture1.Print "Catch Dot Game Final"

    px = 17 'プレイヤーキャラクタの初期座標
    py = 15
    modec = 0

    pm = 7 'プレイヤーキャラクタの通常の移動速度
    pm2 = 5 'プレイヤーキャラクタのスピードアップ中の移動速度
    pmc = pm

    dotc = 100

    '敵キャラクタ初期化
    For i = 0 To 3
        ef(i) = 1 '存在フラグをたてる
        eh(i) = 0
        em(i) = 9 + i * 2 '敵キャラクタによって速度差を付ける
        emc(i) = em(i)
    Next
    ex(0) = 1 '敵キャラクタの初期座標
    ey(0) = 5
    ex(1) = 33
    ey(1) = 5
    ex(2) = 1
    ey(2) = 25
    ex(3) = 33
    ey(3) = 25

    '最初に壁でゲームフィールドを埋める
    For y = 4 To 26
        For x = 0 To 34
            kg3(x, y) = 1 '壁に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(128, 0, 128), BF '壁表示
        Next
    Next

    '通路を作る
    For y = 5 To 25 Step 4
        For x = 1 To 33
            kg3(x, y) = 0 '空白に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    For x = 1 To 33 Step 4
        For y = 5 To 25
            kg3(x, y) = 0 '空白に対応するコード
             Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    '一部の通路を壁に戻す
    For i = 0 To 14
        f = 0
        While f = 0
            f = 1
            x = Int(Rnd(1) * 9) * 4 + 1
            y = Int(Rnd(1) * 6) * 4 + 5
            h = Int(Rnd(1) * 4)
            c0 = 0
            c1 = 0
            For j = 0 To 3
                If kg3(x + dx(j), y + dy(j)) = 0 Then
                    c0 = c0 + 1
                End If
                If kg3(x + dx(h), y + dy(h)) = 0 Then
                     If kg3(x + dx(h) * 4 + dx(j), y + dy(h) * 4 + dy(j))
= 0 Then
                        c1 = c1 + 1
                    End If
                Else
                    c1 = 0
                End If
            Next
            If c0 < 3 Or c1 < 3 Then
                f = 0
            ElseIf kg3(x + dx(h), y + dy(h)) <> 0 Then
                f = 0
            End If
        Wend
        For j = 1 To 3
            kg3(x + dx(h) * j, y + dy(h) * j) = 1
            xx = x + dx(h) * j
            yy = y + dy(h) * j
              Picture1.Line (xx * 16, yy * 16)-(xx * 16 + 15, yy * 16 +
15), RGB(128, 0, 128), BF '壁表示
        Next
    Next

    'プレイヤーキャラクタの出現場所の周囲は強制的に通路にする
    x = 17
    For y = 13 To 17
        kg3(x, y) = 0
        Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), RGB(0,
0, 0), BF '空白（通路）表示
    Next

    '仮想画面クリア
    For y = 4 To 26
        For x = 0 To 34
            kg1(x, y) = 0 '敵キャラクタが存在しないことを示すコード
            kg2(x, y) = 0 'スピードアップキャラクタが存在しないことを示す
コード
            For i = 0 To 3
                kg4(i, x, y) = 100000 '敵キャラクタ追跡用仮想画面クリア
            Next
        Next
    Next

    'スピードアップキャラクタ初期化
    For i = 0 To 3
        sf(i) = 1 '存在フラグをたてる
        sh(i) = 0
        sm(i) = 10
        smc(i) = sm(i)
        '初期座標を乱数で決定
        x = Int(Rnd(1) * 35)
        y = Int(Rnd(1) * 23) + 4
        While (kg3(x, y) = 1 Or x = px Or y = py)
            x = x + 1 '壁やプレイヤーキャラクタと重なった場合は位置をずら
していく
            If x > 34 Then
                x = 1
                y = y + 1
                If y > 26 Then
                    y = 4
                End If
            End If
        Wend
        sx(i) = x
        sy(i) = y
    Next

    score = 0
    ti = 7000
    putscore

    gamestat = 0 'ゲームモード（０でゲーム中、１でゲームオーバー状態）

    Timer1.Interval = 10 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
     Dim npx%, npy%, npvx%, npvy%, nph1%, nph2%, mf%, nex%, ney%, nsx%,
nsy%
    Dim i%, f!, d%, h%, c%, k0%, k1%, nh%, i0%, i1%, i2%, x%, y%, m%
    Dim col!
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        pmc = pmc - 1 '移動カウンタ更新
        If pmc < 1 Then '移動するか？
            pmc = pm
            If spupc > 0 Then 'スピードアップ中は速度を上げる
                pmc = pm2
                spupc = spupc - 1
            End If
            If modec > 0 Then 'パワーアップ期間を数える
                modec = modec - 1
            End If
            npx = px
            npy = py
            npvx = 0
            npvy = 0
            If keystat(2) = 1 Then '下キーを押しているか？
                npvy = npvy + 1
            End If
            If keystat(4) = 1 Then '左キーを押しているか？
                npvx = npvx - 1
            End If
            If keystat(6) = 1 Then '右キーを押しているか？
                npvx = npvx + 1
            End If
            If keystat(8) = 1 Then '上キーを押しているか？
                npvy = npvy - 1
            End If
            If npvx <> 0 Or npvy <> 0 Then '移動キーを押しているか？
                If npvx <> 0 And npvy <> 0 Then '移動キーを２つ押している
か？
                    If ph = 0 Or ph = 2 Then
                        nph1 = 2 - npvx '第１候補
                        nph2 = 1 + npvy '第２候補
                    Else
                        nph1 = 1 + npvy
                        nph2 = 2 - npvx
                    End If
                Else
                    If npvx <> 0 Then
                        nph1 = 2 - npvx '押されている移動キーが１つの場合
は第１候補と第２候補を同じものにする
                        nph2 = 2 - npvx
                    Else
                        nph1 = 1 + npvy
                        nph2 = 1 + npvy
                    End If
                End If
                 If kg3(npx + dx(nph1), npy + dy(nph1)) <> 1 Then '第１候
補の方向が壁で無いか？
                    mf = 1
                    ph = nph1
                 ElseIf kg3(npx + dx(nph2), npy + dy(nph2)) <> 1 Then '第
２候補の方向が壁で無いか？
                    mf = 1
                    ph = nph2
                Else
                    mf = 0 '第１候補＆第２候補とも壁で移動できない場合
                End If
                If mf = 1 Then '移動できる場合
                    npx = npx + dx(ph) '移動できる方向に移動
                    npy = npy + dy(ph)
```

→つぎのページに続く

リスト1の続き

```
                        putplayer px, py, 0
                        putplayer npx, npy, RGB(255, 255, 255)
                        px = npx '座標更新
                        py = npy
                        If kg3(px, py) = 2 Then '移動先にドットがあるか？
                            kg3(px, py) = 0 '仮想画面のドットを消す
                            putdot px, py, 0
                            scoreup 10 + Int(ti / 100)    '得点アップ
                        End If
                        If kg3(px, py) = 3 Then '移動先にビッグドットがあるか？
                            kg3(px, py) = 0 '仮想画面のビッグドットを消す
                            putbdot px, py, 0
                            scoreup 100
                            modec = modec + 30 + ti / 100
                        End If
                        If kg2(px, py) > 0 Then '移動先にスピードアップキャラクタがあるか？
                            sf(kg2(px, py) - 1) = 2 'キャラクタ消滅処理を指示
                            spupc = spupc + 50 '速度アップ期間追加
                        End If
                        If kg1(px, py) > 0 Then '移動先に敵キャラクタがあるか？
                            If ef(kg1(px, py) - 1) = 1 Then '通常状態の敵キャラクタか？
                                If modec > 0 Then 'パワーアップ中か？
                                     ef(kg1(px, py) - 1) = 3  '敵キャラクタをしばらく消滅させる
                                Else
                                     pf = 1 'プレイヤーキャラクタダメージフラグ
                                    ef(kg1(px, py) - 1) = 2 '敵キャラクタをしばらく硬直させる
                                    emc(kg1(px, py) - 1) = 100
                                End If
                            End If
                        End If
                    End If
                End If
            End If
            If spupc > 0 Then col = RGB(255, 255, 0) Else col = RGB(255, 255, 255)
            If modec > 0 Then col = RGB(0, 255, 255)
            putplayer px, py, col
            '敵キャラクタの処理
            If modec > 0 Then
                col = RGB(0, 255, 0)
                m = 1
            Else
                col = RGB(255, 0, 0)
                m = 0
            End If
            For i = 0 To 3
                If ef(i) = 0 Then '存在していない場合
                ElseIf ef(i) = 1 Then '存在している場合
                    emc(i) = emc(i) - 1 '移動カウンタ更新
                    If emc(i) < 1 Then '移動するか？
                        emc(i) = em(i)
                        i0 = eh(i) 'インデックス１：敵キャラクタ方向
                         i1 = Sgn(px - ex(i)) + 1 'インデックス２：プレイヤーキャラクタとのＸ座標関係
                         i2 = Sgn(py - ey(i)) + 1 'インデックス３：プレイヤーキャラクタとのＹ座標関係
                        c = dp(i0, i1, i2) '位置関係テーブル参照
                        d = -1
                         For h = 0 To 2 '移動方向テーブルによる方向を順番にチェック
                            nh = eh(i) + eht(c, h, m)
                             If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                            k0 = kg3(ex(i) + dx(nh), ey(i) + dy(nh))
                            k1 = kg1(ex(i) + dx(nh), ey(i) + dy(nh))
                             If (k0 = 0 Or k0 >= 2) And k1 = 0 Then     '移動先が空白かドットならば移動可能
                                 If kg4(i, ex(i) + dx(nh), ey(i) + dy(nh)) - ti > 500 Or h = 3 Then '近い過去の間に通過しているかどうか？
                                    d = nh
                                    h = 2
                                End If
                            End If
                        Next
                        If d < 0 Then '移動履歴の影響で進行方向が見つからなかった場合は移動履歴を無視する
                             For h = 0 To 3 '移動方向テーブルによる方向を順番にチェック
                                nh = eh(i) + eht(c, h, m)
                                 If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                                k0 = kg3(ex(i) + dx(nh), ey(i) + dy(nh))
                                k1 = kg1(ex(i) + dx(nh), ey(i) + dy(nh))
                                 If (k0 = 0 Or k0 >= 2) And k1 = 0 Then     '移動先が空白かドットならば移動可能
                                    d = nh
                                    h = 3
                                End If
                            Next
                        End If
                        If d > -1 Then
                            eh(i) = d
                            nex = ex(i) + dx(eh(i)) '移動先の座標
                            ney = ey(i) + dy(eh(i))
                            putenemy ex(i), ey(i), RGB(0, 0, 0)
                            putenemy nex, ney, col
                            kg1(ex(i), ey(i)) = 0
                            kg1(nex, ney) = i + 1
                            If kg3(ex(i), ey(i)) = 2 Then '元の場所にドットがあった場合
                                putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                            End If
                            If kg3(ex(i), ey(i)) = 3 Then '元の場所にビッグドットがあった場合
                                 putbdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                            End If
                            If dotc < 1 Then '一定間隔でドットを発生させる
                                If Rnd(1) < 0.2 Then
                                     putbdot ex(i), ey(i), RGB(255, 255, 0) 'ビッグドットを描く
                                    kg3(ex(i), ey(i)) = 3 '仮想画面にビッグドットを設定
                                Else
                                    putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描く
                                    kg3(ex(i), ey(i)) = 2 '仮想画面にドットを設定
                                End If
                                dotc = 100
                            End If
                            dotc = dotc - 1 'ドット発生カウンタ更新
                            ex(i) = nex
                            ey(i) = ney
                            kg4(i, nex, ney) = ti '残り時間を追跡用仮想画面に記録
                            If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの衝突判定
                                If modec > 0 Then
                                    ef(i) = 3 '敵キャラクタをしばらく消滅させる
                                Else
                                    pf = 1 'プレイヤーダメージフラグ
                                    ef(i) = 2 '敵キャラクタをしばらく硬直させる
                                    emc(i) = 100
                                End If
                            End If
                        End If
                    End If
                ElseIf ef(i) = 2 Then '硬直している場合
                    emc(i) = emc(i) - 1
                    If emc(i) < 1 Then
                        ef(i) = 1 '通常に戻る
                    End If
                    putenemy ex(i), ey(i), RGB(128, 128, 128) 'スピードアップキャラクタなどと重なる場合があるので描画し直す
                ElseIf ef(i) = 3 Then '消滅開始
                    kg1(ex(i), ey(i)) = 0 '仮想画面から消去
                    scoreup 100 '得点アップ
                    ef(i) = 4 '消滅状態へ移行
                ElseIf ef(i) = 4 Then '消滅中
                    putenemy ex(i), ey(i), RGB(128, 128, 128)
                    If modec < 1 Then 'パワーアップが終了したか？
                        ef(i) = 1 '通常状態へ復帰
                    End If
                End If
            Next
            'スピードアップキャラクタの処理
            For i = 0 To 3
                If sf(i) = 0 Then '存在していない場合
                ElseIf sf(i) = 1 Then '存在している場合
                    smc(i) = smc(i) - 1 '移動カウンタ更新
                    If smc(i) < 1 Then '移動するか？
                        smc(i) = sm(i)
                        c = Int(Rnd(1) * 4) '乱数で移動方向優先順位を決める
                        d = -1
                         For h = 0 To 3 '移動方向テーブルによる方向を順番にチェック
                            nh = sh(i) + eht(c, h, 0)
                             If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                            k0 = kg3(sx(i) + dx(nh), sy(i) + dy(nh))
                            k1 = kg2(sx(i) + dx(nh), sy(i) + dy(nh))
                             If (k0 = 0 Or k0 >= 2) And k1 = 0 Then     '移動先が空白かドットならば移動（敵キャラクタとは重なる場合あり）
                                d = nh
                                h = 3
                            End If
                        Next
                        If d > -1 Then
                            sh(i) = d
                            nsx = sx(i) + dx(sh(i)) '移動先の座標
                            nsy = sy(i) + dy(sh(i))
                            putspup sx(i), sy(i), RGB(0, 0, 0)
                            putspup nsx, nsy, RGB(0, 255, 0)
                            kg2(sx(i), sy(i)) = 0
                            kg2(nsx, nsy) = i + 1
                             If kg3(sx(i), sy(i)) = 2 And kg1(sx(i), sy(i)) = 0 Then '元の場所にドットだけがあった場合
                                putdot sx(i), sy(i), RGB(255, 255, 0) 'ドットを描き直す
                            End If
                             If kg3(sx(i), sy(i)) = 3 And kg1(sx(i), sy(i)) = 0 Then '元の場所にビッグドットだけがあった場合
                                 putbdot sx(i), sy(i), RGB(255, 255, 0) 'ビッグドットを描き直す
                            End If
                            sx(i) = nsx
                            sy(i) = nsy
                            If px = sx(i) And py = sy(i) Then 'プレイヤーキャラクタとの衝突判定
                                spupc = spupc + 50 '速度アップ期間追加
                                sf(i) = 2
                            End If
                        End If
                    End If
                ElseIf sf(i) = 2 Then 'プレイヤーキャラクタと衝突した場合
                    kg2(sx(i), sy(i)) = 0
                    putspup sx(i), sy(i), 0
                    '新しい座標を乱数で決定
                    x = Int(Rnd(1) * 35)
                    y = Int(Rnd(1) * 23) + 4
                    While (kg3(x, y) = 1 Or x = px Or y = py)
                        x = x + 1 '壁やプレイヤーキャラクタと重なった場合は位置をずらしていく
                         If x > 34 Then
                            x = 1
                            y = y + 1
                            If y > 26 Then
                                y = 4
                            End If
                        End If
                    Wend
                    sx(i) = x
                    sy(i) = y
                    sf(i) = 1
                End If
            Next
            If pf > 0 Then 'プレイヤーがダメージを受けている？
                ti = ti - 200 '残り時間を減らす
                pf = 0
            End If
            ti = ti - 1
            If ti < 0 Then ti = 0
            puttime
            If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
        ElseIf gamestat = 1 Then
            With Picture1
                .CurrentX = 230
                .CurrentY = 200
                .FontSize = 15
                .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
            End With
            Picture1.Print "Game Over"
            If keytrig3 = 1 Then
                Timer1.Interval = 0 'インターバルタイマ停止
                gameinit
            End If
        End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Circle (x * 16 + 8, y * 16 + 8), 7, c
End Sub

Private Sub putdot(x, y, c)
    Picture1.Line (x * 16 + 7, y * 16 + 7)-(x * 16 + 8, y * 16 + 8), c, BF
End Sub

Private Sub putenemy(x, y, c)
    Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), c
    Picture1.Line (x * 16 + 15, y * 16)-(x * 16, y * 16 + 15), c
End Sub

Private Sub putspup(x, y, c)
    Picture1.Line (x * 16 + 2, y * 16 + 2)-(x * 16 + 13, y * 16 + 13), c, B
End Sub

Private Sub putbdot(x, y, c)
    Picture1.Line (x * 16 + 5, y * 16 + 5)-(x * 16 + 10, y * 16 + 10), c, BF
End Sub
```

リスト1入力＆実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，[標準EXE]を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真①**のような画面が表示されます。

②フォームの設定

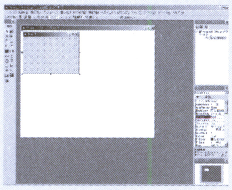
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

③リストの入力

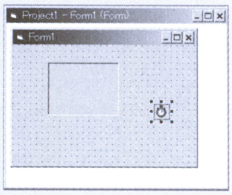
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます。



《写真①》VBを起動したら「標準EXE」を選ぼう



《写真②》PictureBoxとTimerをフォームに配置する

はじめよう Microsoft Excelで BASICプログラミング

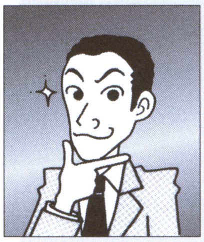
協力 富士通南九州システムエンジニアリング

Excelのマクロがマスターできる!!

第10回 コンピュータ対戦型のパズルゲームを作ろう(3) ～オセロゲーム～

この連載では，Microsoft Excelを使ってBASICのプログラミングを行います。今回からしばらくオセロゲームをテーマに対戦型のボードゲームを作成するテクニックを紹介します。

TEXT & PROGRAMING 椛島たけし



1月号では，CPUの歴史の記事を見て，PCと出会った頃を思い出した。20年前，電器店の店先で8ビットのPC相手に，リストを入力し家に帰る前に，とりあえず動けば満足してたなあ。

ボードゲームに挑戦

先月までは，じゃんけんをベースに，確率的に五分の勝負をいかにコンピュータ有利にするかを紹介しました。今回から一歩進んで，ボードゲームの作成を通して，「プレイヤーの能力で差がつくゲーム」についてプログラミングします。

ボードゲームの中でも，チェスや将棋，囲碁などが「人 対 コンピュータ」や「コンピュータ 対 コンピュータ」としてよく取り上げられます。コンピュータ同士の対戦会も開かれています。ルールが複雑になればなるほどプログラミングがむずかしくなりますので，まずはルールが比較的簡単な「オセロ（リバーシ）」を題材にしてみます。

コンピュータ対戦型オセロゲームに必要な要素

まず，オセロゲームに必要な要素，プログラミングすべき要素を考えてみます。だいたいつぎの三つが必要な要素ですね。

- ●画面表示とユーザー・インターフェース
- ●オセロのルール
- ●コンピュータの手を決める

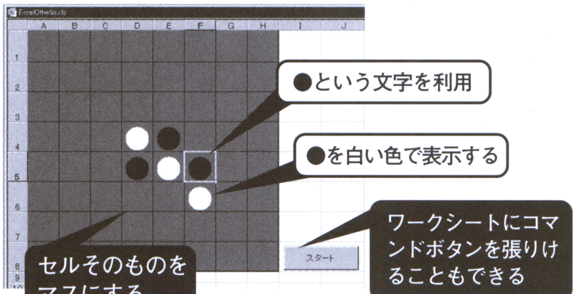
ではそれぞれの要素を，Excel VBAで実現する方法について考えましょう。

「画面表示とユーザー・インターフェース」

オセロ盤の表示には，Excelシートをそのまま利用できます。また，白と黒のコマは文字の●で表し，文字の色を白と黒にすればよさそうです（第1図）。マスに該当するセルをクリックすると自分のコマが表示されるようにすればOKですね。

この段階のポイントはこんな感じです。

- ・背景色やマスの線，文字の大きさは，あらかじめシートにデザインしておく
- ・文字の色，つまりコマの色はプログラム内でFontオブジェクトのColorプロパティを変更する
- ・シートにはクリックイベントがないため，SelectionChangeイベントで代用する



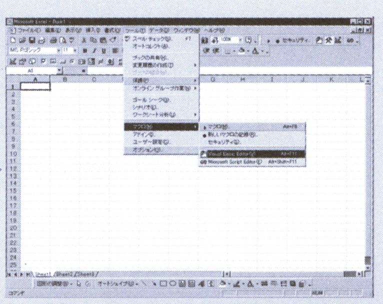
《第1図》Excelのワークシートをオセロ盤に利用する

これでバッチリ！ Excelのプログラム作成ガイド

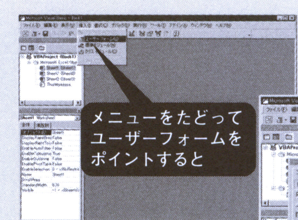
画面デザイン例

Step1

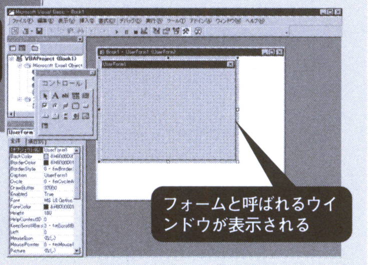
EXCELを起動し，メニューの[ツール][マクロ][Visual Basic Editor]を選択します（第1図）。



《第1図》ExcelでBASICを使うために，Visual Basic Editorを起動する

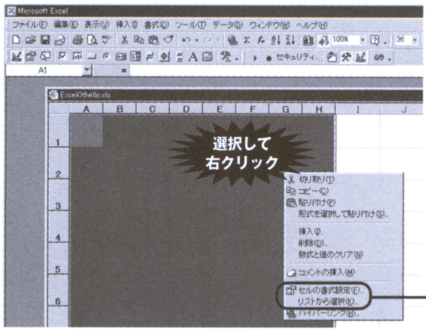


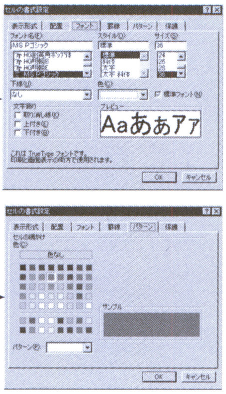
《第2図》Visual Basic Editorが起動されたら，画面デザインのためユーザーフォームを作る



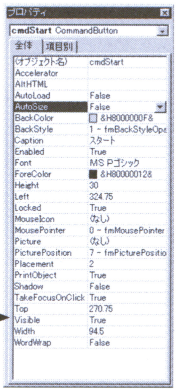
Step2

[挿入]メニューから[ユーザーフォーム]を選択します（第2図）。



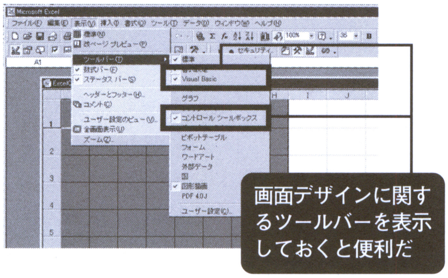


《第2図》セルの書式を選択し，オセロ盤をレイアウトする



《第3図》ワークシートにコントロールを貼り付ける

Excel VBAなどのイベント駆動型プログラミングが便利なのは，アイデア段階で画面デザインを行うことができることです。この段階では第2図，第3図のように，画面にデザインを行いましょう。※補足＝画面デザインの前にツールバーを出しておくと便利です。



(補足)画面デザインの前にツールバーを表示しておこう

「ルール」

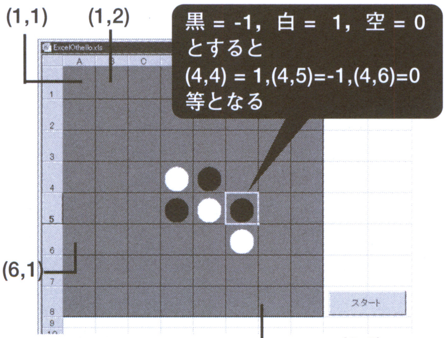
「ルール」については，オセロのルールをそのままプログラミングすればいいわけですね。まず，プログラム時に忘れないように，ルールのリストを作っておくことにしましょう。オセロのルールを箇条書きにすると，第1表のような感じになりますね。

今回は準備段階なので，わかりやすいように，プレイヤーが先手（黒）と限定して進めます。また，盤面の状態については，配列を利用します。この配列がそのまま実際の盤面になるわけです（第4図）。

まずは，ルールを含めた全体のフローチャートを作ってみましょう（第5図）。こうすると実際のプログラミングのイメージがつかめるはずです。

「プレイヤーの手の後の操作」

プレイヤーがクリックしたマスが，ルールに従ってコマを置くことのできるマスか調べます。盤面上であることや，空きのマスであるかは先ほどの配列を調べればすぐわかります。しかし，この段階でもっともむずかしいのが，「相手のコマをはさめる部分にしか自分のコマを置くことができない」というルールです。言葉にすると簡単ですが，プログラミング



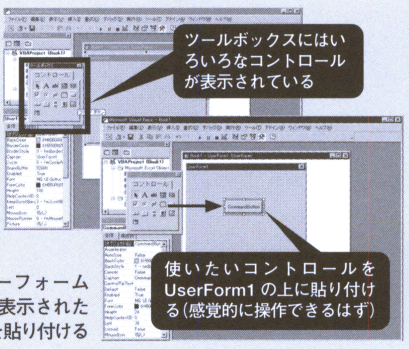
《第4図》盤を表すセルと対応する配列を作る

《第1表》オセロってどんなゲーム？

オセロ・ゲームルール
マスのサイズは8×8の64
中央のマスに白黒それぞれ二つのコマを互い違いに配置する
先手（黒）・後手（白）を決め、互いにコマを置く
空きのマスにしかコマを置くことはできない
相手のコマに隣接し、かつ相手のコマをはさめるマスにしかコマを置くことはできない
はさまれたコマはすべて相手のコマの色となる
上記ルールに従い、コマを置くマスがない場合はパスをする
上記ルールに従い、コマを置くマスがある場合はかならずコマを置かなければならない
上記ルールに従い、互いにコマを置くマスがない場合は終了
終了時にコマの多いほうが勝ち

Step3

ツールボックスのコントロールをクリックしたあと，UserForm1をクリックし，コントロールをUserForm1に貼り付けます（第3図）。

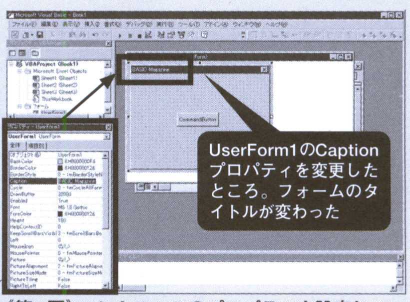


《第3図》ユーザーフォーム(UserForm1）が表示されたら，コントロールを貼り付ける

Step4

画面のデザイン(コントロールの貼り付け)が終わったら，プロパティウインドウでプロパティの設定を行います（第4図）。

プロパティウインドウではプロパティ(属性)と呼ばれるコントロールの設定を行うことができる



《第4図》コントロールのプロパティを設定し，コントロールの性質を決める

するのは意外とむずかしいかもしれません。

第6図の方法で，オセロ盤を表す配列を調べ，コマが置けるか調べます。このような，比較的複雑な処理については，処理ごとのフローチャートを作ると便利です（第7図）。

では，ここまでの処理をプログラミングしてみましょう。ちょっと長くなりましたが，リスト1がここまでのプログラムです。このリストではコンピュータが打ち返さないので，2手で終わってしまいますが，プレイヤーのコマを配置することができるようになりました。つぎに，コンピュータの手を決める段階に進みましょう。

「コンピュータの手を決める」

まず，オセロのルールを覚えたばかりのプレイヤーがどのように上手くなっていくか考えてみましょう。大体，つぎのようなステップで上達していくのではないでしょうか？

①置けるところに置く
②最も多く相手のコマを取れる場所に置く
③端，辺といった一般的に有利な場所を優先して置く

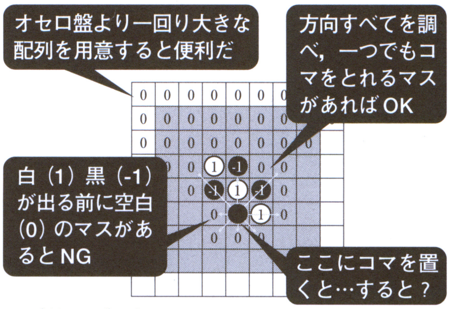
まずは「置けるところに置く」

人間と同じく，コンピュータもまずは一歩一歩，成長させることにして，まずは「置けるところに置く」プログラムを考えてみます。

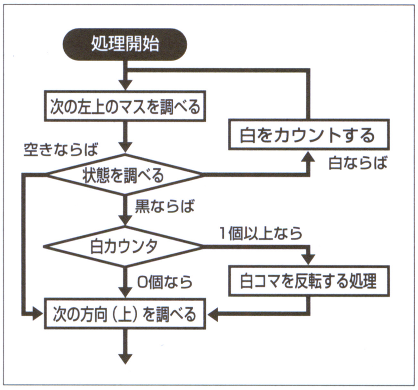
①空きのマスすべてを調べ，手(置ける場所）をすべて見つける
②手の中から，ランダムにコマを置く場所を決める。

まず，①の方法についてはプレイヤーがコマを置く場合とほぼ逆の処理になります。ですから，同じやりかたでOKですね。ただし，手を保存しておく必要がありますので，「ユーザー定義型配列」といったちょっと特殊な変数を使います。

②の，手をランダムに決定するためには，Excel VBAのrnd関数を利用します。だたし，コマを置くことのできる場所の数が毎回変わりますので，注意する必要がありますね。この方法でプログラミングしたリスト（変更・追加分）がリスト2です。ここまでできたら，さっそく実行してみましょう。



《第6図》プレイヤーのクリックしたマスがコマの置けるマスか調べる方法



《第7図》複雑な処理は小さな処理単位でフローチャートをまとめる

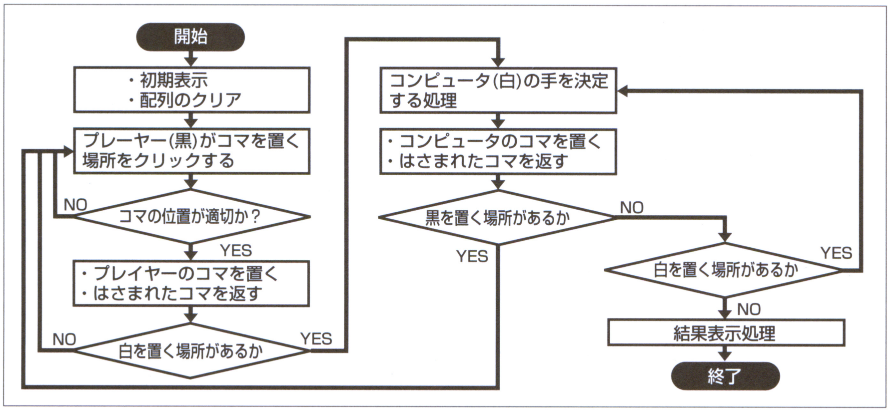
実際対戦してみると，コンピュータはかなり弱いのですが，ときどき予想外の好手を見せてくれます(笑)。これをベースに改良していくことにしますが，今回はヒントだけ紹介します。

最も多く相手のコマを取れる場所

コマを取れる数を計算する必要がありますが，ランダムの場合のプログラムを多少修正するだけでOKです。

一般的に有利なマスを優先する

あらかじめ一般的に有利なマス（角など）と不利なマス（角の周囲など）をポイント化しておき，手を決める際に優先順位を決めておくことでプログラミングができます。

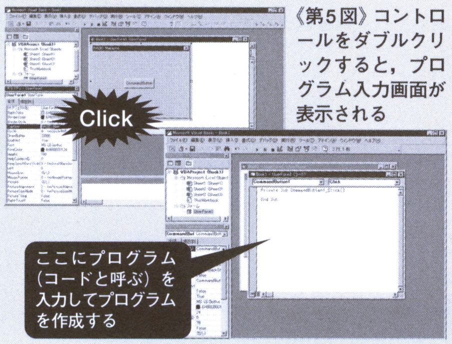


《第5図》ゲームの全体の流れをフローチャートにまとめる

リスト入力方法

Step5

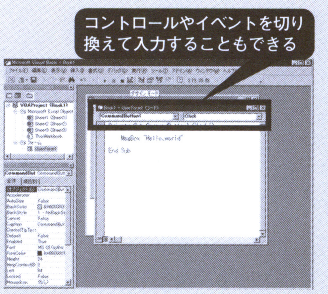
プロジェクトウインドウのUserForm1を右クリックしコードの表示を選ぶ（もしくはコントロールをダブルクリック，コントロールを右クリック後，[コードの表示]を選ぶ)と，プログラムを入力するコードウインドウが表示されます（第5図）。



《第5図》コントロールをダブルクリックすると，プログラム入力画面が表示される

Step6

表示されたコードウインドウにリストを入力します。コントロールとイベントがわかっている場合には，コントロールとイベントを選択して，適切なイベントにリストを入力します（第6図）。



《第6図》コードウインドウにプログラム・リストを入力する

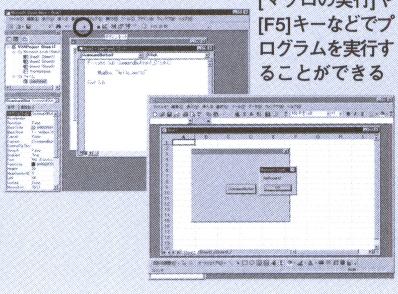
《リスト1》プレイヤーの手を表示するプログラム・リスト

```
Option Explicit

Private Const PIECE_CHR = "●" 'コマを表す文字
Private Const COL_BLACK = &H0       'コマの色(黒)
Private Const COL_WHITE = &HFFFF    'コマの色(白)
Private Const VAL_BLACK = -1     '黒の値
Private Const VAL_WHITE = 1      '白の値
Private Const VAL_SPACE = 0      '空の値

Private Const BOARD_TOP = 1      '盤の上位置
Private Const BOARD_LEFT = 1     '盤の左位置
Private Const BOARD_SIZE = 8     '盤のサイズ

Private Const ERR_OUTOF = 2&     'エラー(盤外)
Private Const ERR_SAME = 4&      'エラー(同じマス)
Private Const ERR_MISS = 8&      'エラー(打てないマス)

'オセロ盤の状況保存用配列
Dim BoardStatus(0 To 9, 0 To 9) As Long
Dim plngCompColor As Long   'コンピューターのコマの色
Dim plngPlayerColor As Long 'プレイヤーのコマの色


'スタートボタンが押された時の処理
Private Sub cmdStart_Click()
    Dim lngRowCtr As Long
    Dim lngColCtr As Long

  'プレイヤーが黒、先手とする
    plngPlayerColor = VAL_BLACK
    plngCompColor = VAL_WHITE

    '状態配列をクリアする
    For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            BoardStatus(lngRowCtr, lngColCtr) = 0
        Next lngColCtr
    Next lngRowCtr

    '初期のコマの配置を行う
    BoardStatus(4, 4) = VAL_WHITE
    BoardStatus(5, 5) = VAL_WHITE
    BoardStatus(4, 5) = VAL_BLACK
    BoardStatus(5, 4) = VAL_BLACK

    '画面表示処理を呼び出す
    Call procDrawPiece

    '一旦盤の外を選択する
    Cells(BOARD_TOP + BOARD_SIZE, BOARD_LEFT + BOARD_SIZE).Activate
End Sub

'ワークシートの選択セルが変更された時
'≒マウスがクリックされたとき
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim lngClickRow As Long 'クリックされた行
    Dim lngClickCol As Long 'クリックされた列
    Dim lngRetCode As Long  'プロシージャの戻り値

    'プレイヤーのコマのチェック処理の呼び出し
    lngRetCode = procPlayerCheck(Target(1, 1).Row, Target(1, 1).Column)
    If lngRetCode = -1 Then  '戻り値が-1(正しいマスを選択した場合)
        BoardStatus(Target(1, 1).Row, Target(1, 1).Column) = -1
        Call procDrawPiece  'コマを表示する処理
    End If
End Sub

'コマを表示する処理
Private Sub procDrawPiece()
    Dim lngRowCtr As Long '行カウンタ
    Dim lngColCtr As Long '列カウンタ

    '配列のすべてを調べる
    For lngColCtr = 1 To 8
        For lngRowCtr = 1 To 8
            Select Case BoardStatus(lngRowCtr, lngColCtr)
            Case VAL_SPACE  '空白なら
                Cells(lngRowCtr, lngColCtr).Value = ""
            Case VAL_BLACK  '黒なら
                'セルの文字色に黒を設定し、●を表示する
                Cells(lngRowCtr, lngColCtr).Font.Color = COL_BLACK
                Cells(lngRowCtr, lngColCtr).Value = PIECE_CHR
            Case VAL_WHITE  '白なら
                'セルの文字色に白を設定し、●を表示する
                Cells(lngRowCtr, lngColCtr).Font.Color = COL_WHITE
                Cells(lngRowCtr, lngColCtr).Value = PIECE_CHR
            End Select
        Next lngRowCtr
    Next lngColCtr

End Sub

'クリックされたマスについて調べる処理
Private Function procPlayerCheck(lngRow As Long, lngCol As Long) As Long
    Dim lngRowCtr As Long
    Dim lngColCtr As Long
    Dim lngReverse As Long
    Dim lngRetRev As Long
    Dim lngRetCode As Long

    'マス範囲外のチェック
    procPlayerCheck = 0
    Select Case lngRow
    Case BOARD_TOP To BOARD_TOP + BOARD_SIZE - 1
        Select Case lngCol
        Case BOARD_LEFT To BOARD_LEFT + BOARD_SIZE - 1
        'すでにコマが無いかチェック
            If BoardStatus(lngRow, lngCol) <> VAL_SPACE Then
                procPlayerCheck = ERR_SAME
            End If
            procPlayerCheck = -1
        Case Else
            procPlayerCheck = ERR_OUTOF
        End Select
    Case Else
        procPlayerCheck = ERR_OUTOF
    End Select

    If procPlayerCheck <> -1 Then Exit Function

    'コマを返せるか調べる
    lngRetCode = procReverseCheck(plngPlayerColor, lngRow, lngCol)
    If lngRetCode > 0 Then 'コマを返せるならば
        'コマを返す処理
        Call procReverse(plngPlayerColor, lngRow, lngCol)
        procPlayerCheck = -1
    Else
        procPlayerCheck = ERR_MISS
    End If


End Function

'打つ手があるか調べる
Private Function procIdeaCheck(CheckColor As Long) As Long
    Dim lngRowCtr As Long '行カウンタ
    Dim lngColCtr As Long '列カウンタ
    Dim lngIdeaCtr As Long '手数のカウンタ
    Dim lngCompIdea As Long
    Dim lngRetCode As Long
    'コマを置ける位置の確認
    For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            '空白のマスの場合
            If BoardStatus(lngRowCtr, lngColCtr) = VAL_SPACE Then
                'コマを返せるか調べる
```

プログラム実行方法

Step7

[実行]メニューから[sub/ユーザフォームの実行]を選びます。または，メニューの下の三角のボタンをクリックしたり，F5キーを押してもプログラムを実行することができます（第7図）。



《第7図》[実行]-[マクロの実行]や[F5]キーなどでプログラムを実行することができる

Excelブックを開くと同時にプログラムを実行する

Point1

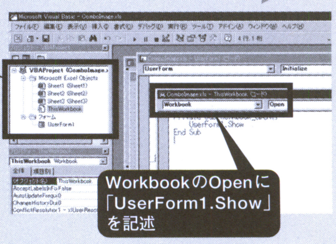
WorkbookのOpenイベントで開きたいフォームのShowメソッドを実行すると，Excelブックを開くと同時にプログラムを実行することができます（第8図）。

(例)

Private Sub Workbook_Open ()

　UserForm1.Show

End Sub

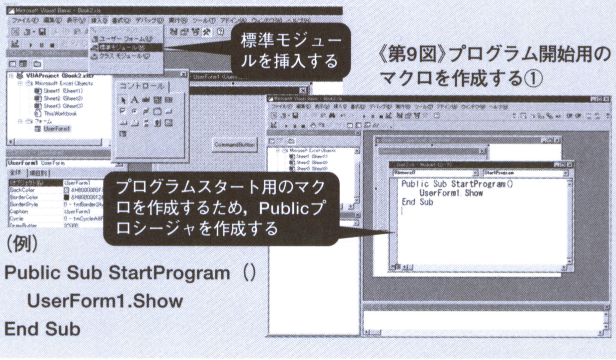


《第8図》ワークブックを開いたときに，プログラムを実行する方法

```
                lngRetCode = procReverseCheck(CheckColor, lngRowCtr, lngColCtr)
                If lngRetCode > 0 Then
                    lngIdeaCtr = lngIdeaCtr + 1
                End If
            End If
        Next lngColCtr
    Next lngRowCtr

    '打てる手の数を返す
    procIdeaCheck = lngIdeaCtr

End Function

'指定されたマスにコマを置けるか調べる
Private Function procReverseCheck(CheckColor As Long, NowRow As Long, NowCol As Long)
As Long
    Dim lngReverse As Long  '返すコマの総数
    Dim lngRetCode As Long  '返すコマの数

    lngReverse = 0
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, -1) '左上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, 0) '上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, 1) '右上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 0, -1) '左
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 0, 1) '右
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, -1) '左下
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, 0) '下
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, 1) '右下
    lngReverse = lngReverse + lngRetCode

    procReverseCheck = lngReverse
End Function

'指定された方向について、返すコマの数を調べる
Private Function procCellsCheck(CheckColor As Long, _
                                NowRow As Long, _
                                NowCol As Long, _
                                RowStep As Long, _
                                ColStep As Long) As Long
    Dim lngTempRow As Long  '一時行カウンタ
    Dim lngTempCol As Long  '一時列カウンタ
    Dim lngRevCtr  As Long  '返せるコマの数
    Dim lngReverseValue As Long
    Dim lngSameValue As Long

    Select Case CheckColor  '調べる色の種類
    Case VAL_BLACK  '黒の場合
        lngSameValue = VAL_BLACK     '同じ色は黒
        lngReverseValue = VAL_WHITE '返す色は白
    Case VAL_WHITE  '白の場合
        lngSameValue = VAL_WHITE     '同じ色は白
        lngReverseValue = VAL_BLACK '返す色は黒
    End Select

    lngRevCtr = 0
    lngTempRow = NowRow
    lngTempCol = NowCol

    Do
        lngTempRow = lngTempRow + RowStep
        lngTempCol = lngTempCol + ColStep
        Select Case BoardStatus(lngTempRow, lngTempCol)
        Case lngReverseValue '置くコマと違う色の場合
            lngRevCtr = lngRevCtr + 1
        Case lngSameValue   '置くコマと同じ色の場合
            If lngRevCtr > 0 Then   'カウントアップ
                procCellsCheck = lngRevCtr
            End If
            Exit Do
        Case VAL_SPACE  '空白の場合
            Exit Do
        End Select
    Loop

End Function


'指定されたマスにコマを置き、コマを返す
Private Sub procReverse(RevColor As Long, NowRow As Long, NowCol As Long)
    Dim lngTempRow As Long  '一時行カウンタ
    Dim lngTempCol As Long  '一時列カウンタ
    Dim lngTempCtr As Long  '一時カウンタ
    Dim lngRevCtr  As Long  '反転カウンタ
    Dim lngRowStep As Long  '行方向
    Dim lngColStep As Long  '列方向
    Dim lngSameValue As Long    '置いたコマと同じ色
    Dim lngReverseValue As Long '返すコマの色

    Select Case RevColor  '色の種類
    Case VAL_BLACK  '黒の場合
        lngSameValue = VAL_BLACK     '同じ色は黒
        lngReverseValue = VAL_WHITE '返す色は白
    Case VAL_WHITE  '白の場合
        lngSameValue = VAL_WHITE     '同じ色は白
        lngReverseValue = VAL_BLACK '返す色は黒
    End Select


    '指定されたマスにコマを置く
    BoardStatus(NowRow, NowCol) = RevColor
    Call procDrawPiece

    If RevColor = plngPlayerColor Then
        MsgBox "プレイヤーの手は" & CStr(NowRow) & "行  " & CStr(NowCol) & "列です。"
    Else
        MsgBox "コンピューターの手は" & CStr(NowRow) & "行  " & CStr(NowCol) & "列で
す。"
    End If

    '8方向について返すコマを調べる
    For lngRowStep = -1 To 1
        For lngColStep = -1 To 1
            lngRevCtr = 0
            lngTempRow = NowRow
            lngTempCol = NowCol

            Do
                lngTempRow = lngTempRow + lngRowStep
                lngTempCol = lngTempCol + lngColStep
                Select Case BoardStatus(lngTempRow, lngTempCol)
                Case lngReverseValue '違う色の場合
                    lngRevCtr = lngRevCtr + 1
                Case lngSameValue   '同じ色の場合
                    If lngRevCtr > 0 Then   '返すコマがあったら
                        'コマを返す
                        For lngTempCtr = 1 To lngRevCtr
                            BoardStatus(NowRow + lngRowStep * lngTempCtr, _
                                    NowCol + lngColStep * lngTempCtr) = lngSameValue
                        Next lngTempCtr
                    End If
                    Exit Do
                Case VAL_SPACE '空白の場合
                    Exit Do
                End Select
            Loop
        Next lngColStep
    Next lngRowStep

End Sub
```

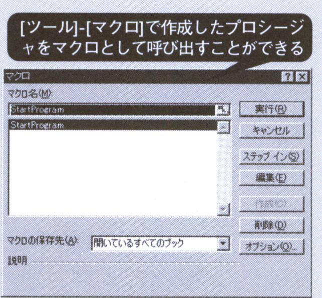
マクロに登録する方法
Point2

標準モジュールにPublicプロシージャを登録すると，ワークシートからマクロとして呼び出すことができます（第9図）。このような設定をすると，わざわざVisual Basic Editorを起動しないでも，[ツール]メニューの[マクロ]でプログラムをスタートするプロシージャを呼び出すことができます（第10図）。



（例）
Public Sub StartProgram ()
UserForm1.Show
End Sub

《第9図》プログラム開始用のマクロを作成する①



《第10図》プログラム開始用のマクロを作成する②

★本ページのプログラムはhttp://www.basicmagazine.netからダウンロードできます。

《リスト2》コンピュータの手を決定するプログラム・リスト（リスト１の変更・追加部分のみ）

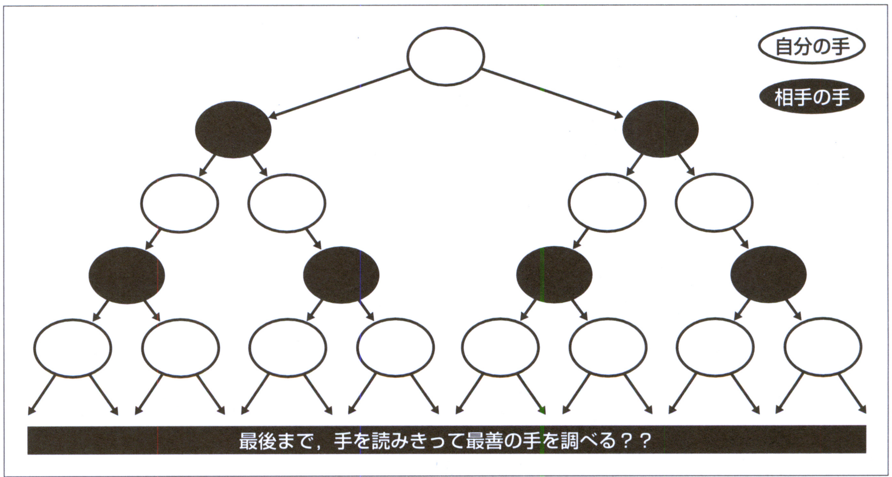
```
'コンピューターの手の格納用のユーザ定義型
Private Type typIdea
    Row As Long
    Col As Long
    Rev As Long
End Type
Dim udtIdea(60) As typIdea


'ワークシートの選択セルが変更された時
'≒マウスがクリックされたとき
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim lngClickRow As Long 'クリックされた行
    Dim lngClickCol As Long 'クリックされた列
    Dim lngRetCode As Long  'プロシージャの戻り値
    Dim boolComp As Boolean
    Dim boolPlayer As Boolean

    Dim lngPlayer As Long  'プロシージャの戻り値
    Dim lngComp As Long  'プロシージャの戻り値

    'プレイヤーのコマのチェック処理の呼び出し
    lngRetCode = procPlayerCheck(Target(1, 1).Row, Target(1, 1).Column)
    If lngRetCode = -1 Then  '戻り値が-1(正しいマスを選択した場合)
        BoardStatus(Target(1, 1).Row, Target(1, 1).Column) = -1
        Call procDrawPiece  'コマを表示する処理


        Do Until lngPlayer > 0
            'コンピューターの打てる場所のチェック
            lngComp = procIdeaCheck(plngCompColor)
            If lngComp > 0 Then  '打てる場所がある場合
                'コンピュータの手を決める処理
                Call procCompIdea(lngComp)
            End If

            lngPlayer = procIdeaCheck(plngPlayerColor)

            If lngPlayer = 0 And lngComp = 0 Then
                MsgBox "終了です"
                Exit Do
            ElseIf lngPlayer = 0 Then
                MsgBox "プレイヤーは打つ手がありません。コンピューターの番です。"
            ElseIf lngComp = 0 Then
                MsgBox "コンピューターは打つ手がありません。あなたの番です。"
            Else
            End If
        Loop
    End If

End Sub


'打つ手があるか調べる
Private Function procIdeaCheck(CheckColor As Long) As Long
    Dim lngRowCtr As Long '行カウンタ
    Dim lngColCtr As Long '列カウンタ
    Dim lngIdeaCtr As Long '手数のカウンタ
    Dim lngCompIdea As Long
    Dim lngRetCode As Long
    'コマを置ける位置の確認
    For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            '空白のマスの場合
            If BoardStatus(lngRowCtr, lngColCtr) = VAL_SPACE Then
                'コマを返せるか調べる
                lngRetCode = procReverseCheck(CheckColor, lngRowCtr, lngColCtr)
                If lngRetCode > 0 Then
                    lngIdeaCtr = lngIdeaCtr + 1
                    With udtIdea(lngIdeaCtr)
                        .Row = lngRowCtr
                        .Col = lngColCtr
                        .Rev = lngRetCode
                    End With
                End If
            End If
        Next lngColCtr
    Next lngRowCtr

    procIdeaCheck = lngIdeaCtr

End Function

'コンピュータの手を決める処理
Private Sub procCompIdea(IdeaCount As Long)
    Dim lngCompIdea As Long

    lngCompIdea = Int((IdeaCount * Rnd) + 1)
    Call procReverse(plngCompColor, udtIdea(lngCompIdea).Row,
udtIdea(lngCompIdea).Col)
    Call procDrawPiece

End Sub

'
```

先を読む

みなさんはもう気付いていると思いますが，今までの方法は重大なことを見落としています。つぎの相手の手のことを考えていませんね。いわば相手のことを考えず，自分の打ちたいところに打つやりかたでした。人と人の対戦のように，自分（コンピュータ）の手が，相手（プレイヤー）にとって有利か不利か調べることも必要です。

この「相手の手を読む」ことをプログラミングするのが最も大変なのです。ちょっと考えてみましょう。

もちろん，自分（相手）にとって何が有利か不利かは，すべての手を判定し調べないとわかりません。一見，不利な手が後から有利な手にかわることもありますよね。まず，すべての手を読みきることを考えてみましょう（第８図）。



《第８図》可能性のある「手」をすべて判断しようとすると・・・

オセロの盤面は８×８の64マスです。最初から4マス使用されていますので，コマを置く回数が60回あります。仮に平均二つの手（選択肢）があるとすると，最終的な手の数は2の60乗です。この手を評価するために必要な時間をちょっと割り出してみましょう。「最近のパソコンは速いからそのぐらいの計算すぐなんじゃないの？」と思うかもしれませんが･･･。

超高性能なパソコンで１秒間に１億とおりの手の判断ができるとすると，すべての手の判断にかかる時間は2^60 ÷ 1億（回）÷ 60（秒）÷ 60（分）÷ 24（時間）÷ 365（日）≒ 365.589･･･（年）となってしまいます。

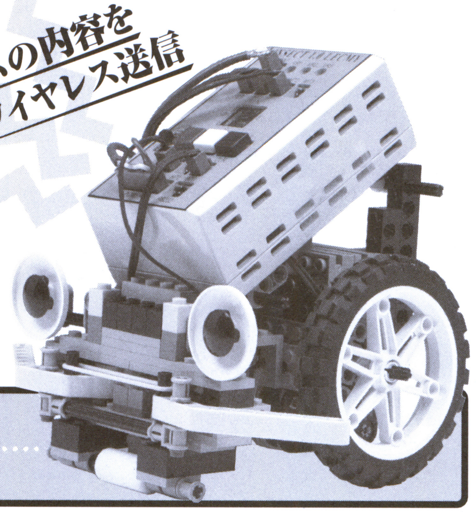
実際に選択肢は二つ程度ではありませんから，もっと大変なことになってしまいます。したがって，先の手を読む場合，読む手の数をうまく減らしていく必要がありますね。

次回は，コンピュータの手を決定する方法をさらに進歩させることにします。お楽しみに。

ナビゲータ搭載の思考型自走ロボット

「INSECT（インセクト）」のすべて

プログラムの内容を磁気でワイヤレス送信



連載1回目

山崎　誠

「INSECT」と他ロボットとのちがい

動作のプログラムができる，自走ロボットでは，LEGO（レゴ）社の「MINDSTORMS（マインドストーム）」（**写真1**）が最も有名です。これはロボット自作ブームのきっかけになった，多くの人に楽しく使われている素晴らしい製品です。

しかし，ユーザーの対象年齢を低く設定しているためか，高度で知的な動きをロボットにプログラミングするには，少し物足りなさを感じます。

MINDSTORMSの行動は，センサによる情報のフィードバックも含まれてはいますが，メインはタイマ（時間）設定に依存していて，精密な動きをさせようとすると，不満が残ります。

とくに，ロボット自身が，位置情報を持たないので，基本的にはあいまいな運転になってしまいます。

そこで筆者が考えたのが，ナビゲーション搭載の自走ロボット・システム「INSECT（インセクト）」です（**写真2～5**）。

簡潔に言うと「自身の位置情報を持って正確に動くロボット」ということになります。

センサ入力3チャンネル，モータ・ドライブ3チャンネルで，自走ロボットが実現できました。

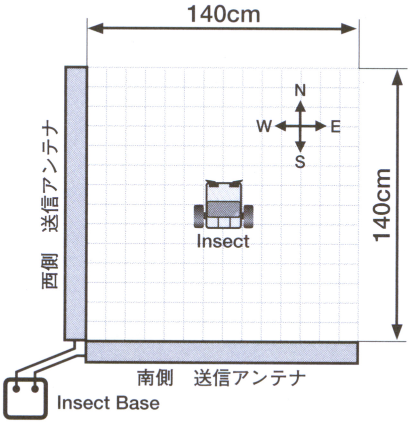
「INSECT（昆虫）」命名の由来は，筆者には5歳の息子がおり，この子がロボットの動きを見て「虫みたい」と言ったことがきっかけです。

INSECTの動作原理

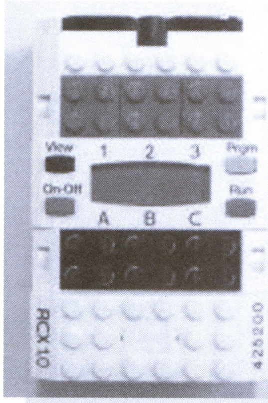
二つの送信アンテナを配置した範囲であれば，INSECTを搭載したロボットは，自身の位置をX-Y座標として取得します。

送信アンテナからは磁気信号が出ていて，これをINSECTは検出して位置を割り出します。

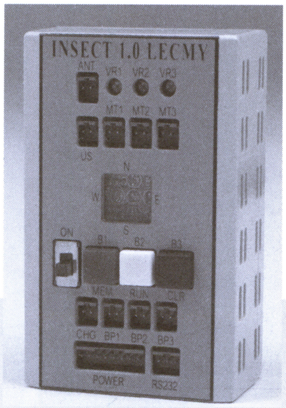
第1図のように，見えない磁気の線が



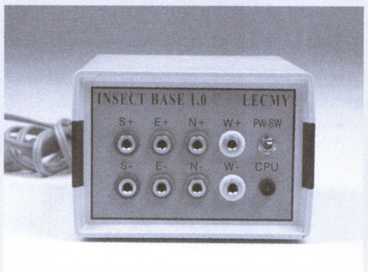
《第1図》2基のアンテナが作り出す正確な位置情報が本システムの特徴



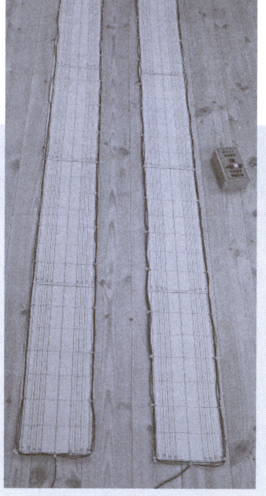
《写真1》プログラム可能なロボットとして大人気のマインドストームの心臓部「RCX」



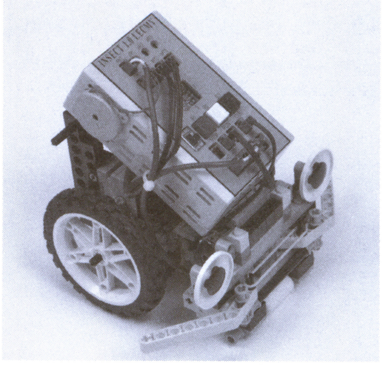
《写真2》INSECTの本体部



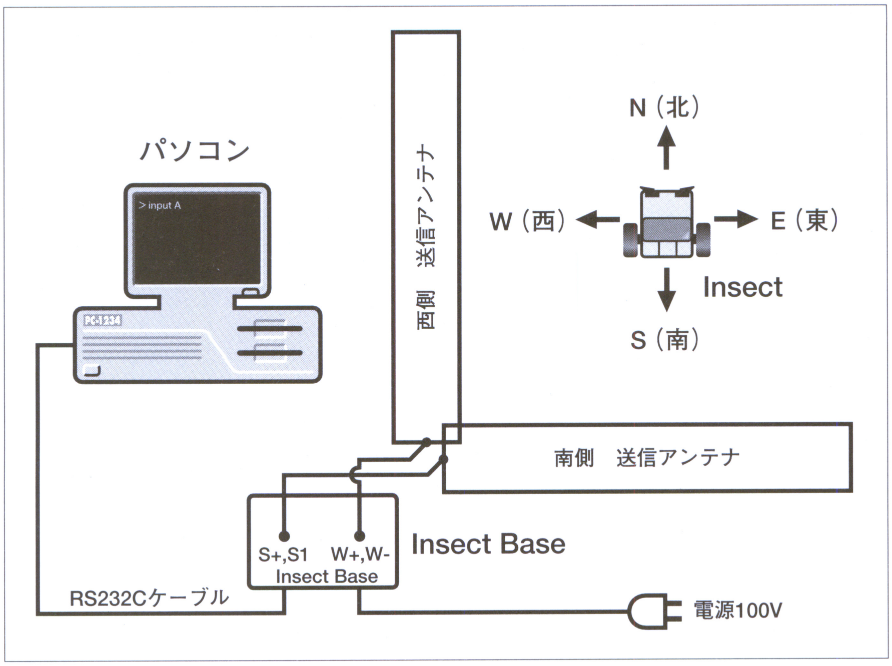
《写真3》パソコンからのデータを送るInsect Base



《写真4》磁気の乗せたデータを送信するアンテナ



《写真5》INSECTを積んだ自走ロボットの例



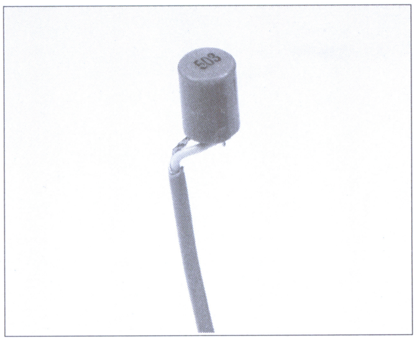
《第2図》全体の構成図

縦横に走っているとイメージすると，わかりやすいでしょう。

INSECTは，これを検出して走行します。

そのため，途中で複雑な行動をしたとしても，最初の出発地に帰すことも可能です。

線の数は，縦1024本，横1024本ですが，プログラム作成を簡単にするために，縦140本，横140本に制限しています。



《写真6》受信用のアンテナ

ただし内部プロセスは1024本で作動しています。

INSECTの構成

INSECTの構成を第2図に示します。INSECT本体には受信アンテナがあり，送信アンテナからのナビゲーション情報を受信して，現在位置を把握します。

二つの送信アンテナのエリア内で，INSECTを搭載した自走ロボットは動きます。

方位は地球的な方位ではなく，送信アンテナの設置方向で決定されます。

パソコンからは「INSECT Base」に行動プログラムが伝えられ，それが送信アンテナからワイヤレスで送られ，**写真6**の受信アンテナで受けてINSECTの行動プログラムを書き換えます。

行動プログラムをパソコン上で作成してワイヤレスで転送するのは，MINDSTORMSと同じですが，MINDSTORMSは赤外線で転送していますが，筆者が考案した「INSECT」は磁気にて転送します。

送信アンテナからプログラムは磁気の変化となって送出されます。

赤外線は光なので指向性が強く，MINDSTORMSのRCXの方向に正確に当てなければなりません。

しかし磁気は指向性が弱く，ロボットが反対の方向を向いていても，転送できる特徴があります。

INSECTのRUNボタンを押すと，行動プログラムに従って，自走します。パソコンがワイヤレスで動かしているのではありません！

INSECTは書き込まれた行動プログラムで，自身が位置を確認しながら自走するのです。

動作設定をする統合開発環境「INSECT for Windows」

INSECTを十分に使いこなすには，かなりしっかりとした統合開発環境が必要と思い，また，MINDSTORMSを意識して，DirectXを駆使して，きれいに仕上げたソフトを開発しました（**写真7，8**）。

画面中央に表示されるコア（6面体）に，プログラムを行います。処理を簡潔に記述できるように制作しました。各6面にはそれぞれ意味があり，次号で解説していきます。

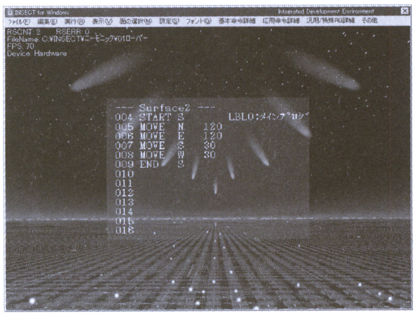
第1表が，INSECT for Windowsを実行させるために必要な最小システムです。

コアへの書き込みは**写真9**の専用エディタで行うスタイルを取りました。

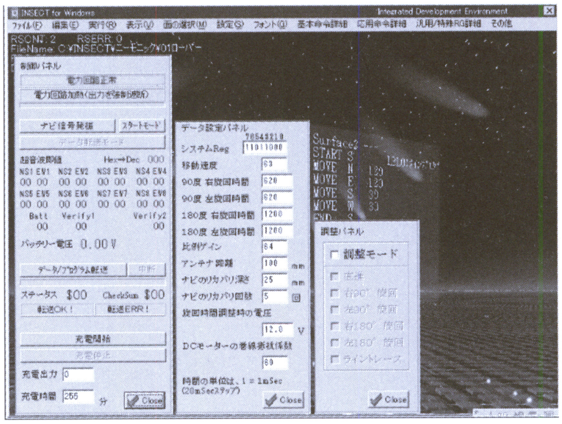
統合開発環境「INSECT for Windows」にこの専用エディタは入っています。

コアに書き込まれたプログラムは，アセンブルして機械語に翻訳してから転送します。

翻訳エンジンは，本プログラムに入っています。



《写真7》Windows搭載パソコンで動く統合開発環境。中央部が六面体のコア



《写真8》制御，データ設定，調整用のパネル

F9キーを押すと，アセンブルが始まります（**写真10**）。

アセンブルが終わったら「データ／プログラム転送」ボタンを押して転送します（**写真11**）。

プログラムは，磁気（ワイヤレス）でINSECTに転送されます。

INSECTのRUNボタン**写真12**を押すと，プログラムに従った動作を行います。

何ができるか（自走編）

さてこのへんでINSECTを動かしてみましょう。

コアは6面あり，それぞれに機能が割り当てられています。Surface2がメイン処理部です（**写真13**）。

マウスでコアをつかんでクルクルと回転させると，各Surfaceのプログラムを確認できます。

写真13のプログラムは簡単なものです。

1：北へ120ポイント走行

2：東へ120ポイント走行

3：南へ30ポイント走行

4：西へ30ポイント走行

第3図のように，四角に走行するプログラムです。

送信アンテナからナビされているので，正確に走行します。

見えない白線を敷いてライン・トレースしているように実に正確です（**写真14**）。

プログラムは，ワイヤレスでINSECTに送信されます。

パソコンで作成されたプログラムは，INSECT Baseに転送されて，送信アンテナから磁気となって，INSECTの受信アンテナに届けられる仕組みです。

何ができるか（フォークリフト工作セット編）

写真15はタミヤ社のフォークリフト工作セット（No.115）にINSECTを組み込んだ例です。

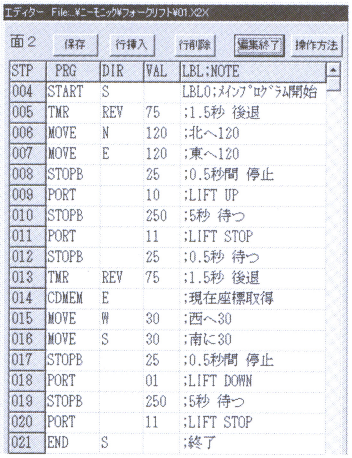
第4図のように，A地点からスタートし，B地点に置いてある荷物を拾い，C地点で荷物を降ろすプログラムも簡単に作成できます。プログラムに慣れてくると，さらに複雑な動きをプログラミングできます。

タミヤのフォークリフト工作セット（**写真16**）は，有線リモコン式で，これだけでも十分に楽しめるキットですが，1時間も遊ぶと飽きてしまいます。

しかし，このキットの良い点は，2輪独立モータで走行でき，リフト上昇下降の機構がしっかりしている点にあります。また小売価格3,000円と安いのも評価の一つです。

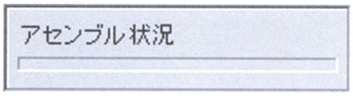
《第1表》INSECT for Windowsを動かすパソコンに要求される性能

DOS/V機専用		
	OS	Windows98SE／Me／2000
	CPU	セレロン300MHz以上，Pentium Ⅲ相当を推奨
	RAM	64MB
	使用ハードディスク容量	100MB
	9ピンシリアルポート	1ch
	サウンド	SoundBlaster16Windows互換サウンドデバイス
	CD-ROM	8倍速
	ビデオ・ディスプレイ	800×600 SVGAビデオ，1MB RAM，カラー16bitカラー

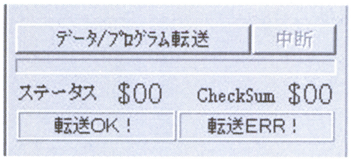


STP	PRG	DIR	VAL	LBL;NOTE
004	START	S		LBL0;ﾒｲﾝﾌﾟﾛｸﾞﾗﾑ開始
005	TMR	REV	75	;1.5秒 後退
006	MOVE	N	120	;北へ120
007	MOVE	E	120	;東へ120
008	STOPB		25	;0.5秒間 停止
009	PORT		10	;LIFT UP
010	STOPB		250	;5秒 待つ
011	PORT		11	;LIFT STOP
012	STOPB		25	;0.5秒 待つ
013	TMR	REV	75	;1.5秒 後退
014	CDMEM	E		;現在座標取得
015	MOVE	W	30	;西へ30
016	MOVE	S	30	;南に30
017	STOPB		25	;0.5秒間 停止
018	PORT		01	;LIFT DOWN
019	STOPB		250	;5秒 待つ
020	PORT		11	;LIFT STOP
021	END	S		;終了

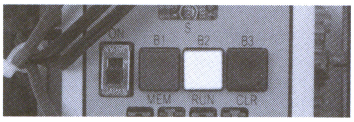
《写真9》コアへの書き込みは専用のエディタで行う



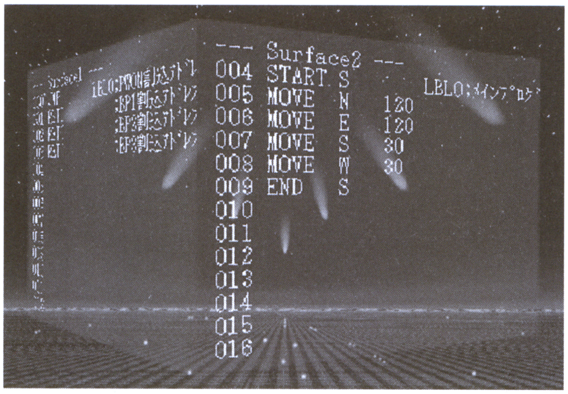
《写真10》アセンブル進行状態を示すメータ



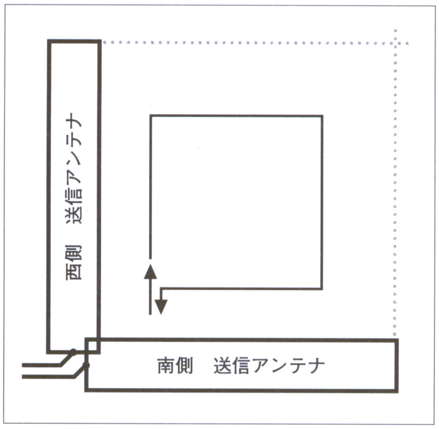
《写真11》このボタンを押すと，データがワイヤレスで転送される



《写真12》INSECT本体のRUNボタンでロボットは活動を開始する

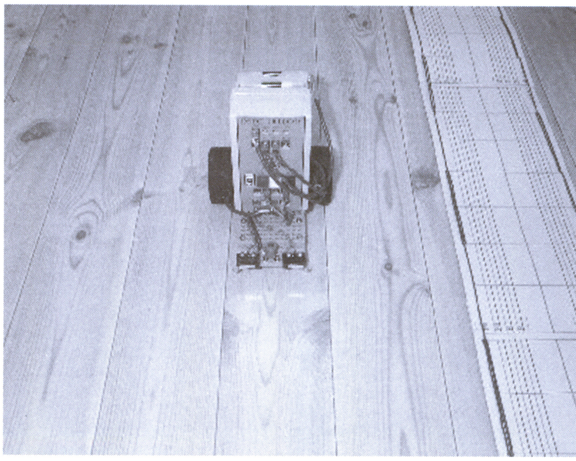


《写真13》自走のプログラム。記述は簡単だ



《第3図》写真13を実行するとロボットはこう動く

《写真14》正確に動作するのはみごと



何ができるか（リモコン操作編）

INSECTは，磁気信号によるワイヤレス・リモコン操作が可能です。

当初は自走がメインなので，リモコンについては考えていなかったのですが，プログラム中に手元に機体を寄せたい時など，パソコン上で遠隔操作できたほうが，行動プログラムの開発が効率的になることから，「リモートモード」を搭載することにしました。

通常は電源が入った時に「自走モード」に固定され，転送されたプログラムに従って動作するのですが，簡単な操作で「リモートモード」に切り替える事ができます。

「リモートモード」への切り替えも，ワイヤレス（磁気信号）で行えます。

INSECT for Windowsを起動し，F8キーを押すと「リモートコントロールパネル」が開きます（**写真17**）。

「リモートモードの実行」ボタンを押すことで，送信アンテナから磁気信号が出力され，INSECTはリモートモードに切り替わります。

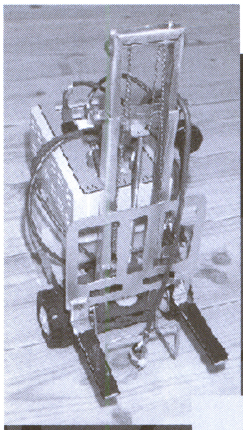
もう一度，リモートモードの実行ボタンを押すと，リモートモードは解除されます（オルタネイト動作です。）

「ジョイステックを使う」にチェックを付けると，アナログ・ジョイスティックが使用可能になります。

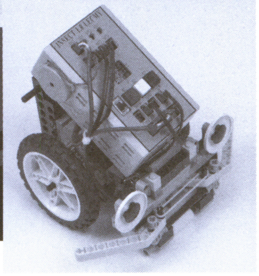
前進，後退，左旋回，右旋回，UP，DOWNの6chをワイヤレスで操作できます。

UP，DOWNはMT3出力の操作になっています。

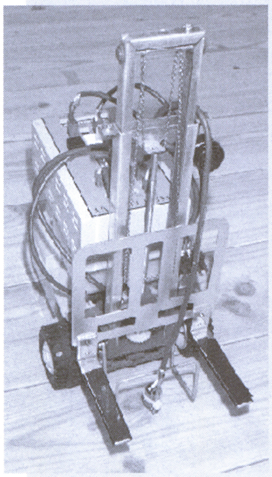
フォークリフトやその他の機体もリモコン操作ができます。



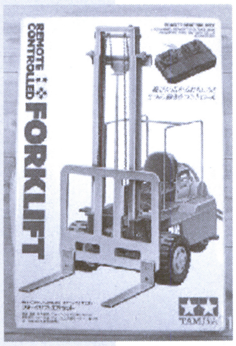
《写真18》リモコン操作でフォーク部も動かせる



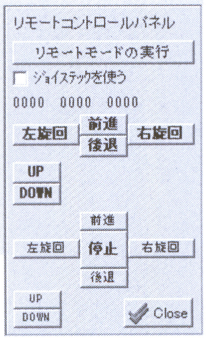
《写真19》レバーの遠隔操作も可能



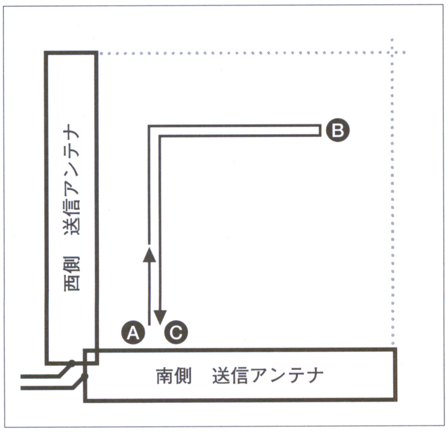
《写真15》フォークリフト工作セットに組み込んだ例



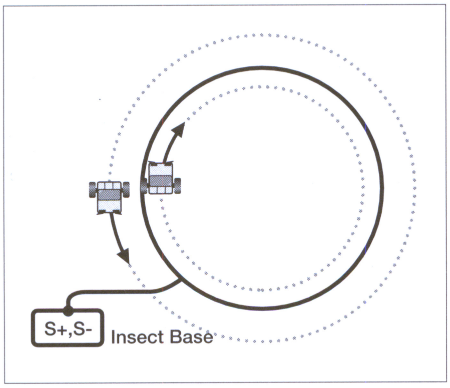
《写真16》しっかりした作りでINSECTとは相性が良い工作セットのパッケージ。模型屋さんやDIY店で買える



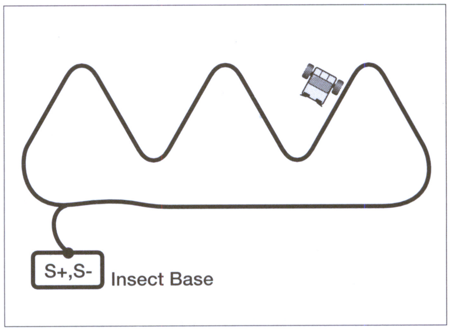
《写真17》PC画面に出てくるリモコン用のパネル



《第4図》2地点を正確に移動させることも可能



《第5図》ライントレースもできる



《第6図》曲線走行も自在

何ができるか（ライントレース編）

INSECTは磁気信号を白線に見たてた，ライントレース・モードが使えます。

従来ライントレースというと床に白線を敷いて，光センサでそれを見ながら走行させるものですが，INSECTは磁気を白線と見なして走行します。

0.2mm角程度の電線（長さ14 m）を，**第5図**のように円形に配置します。調整モードで，ライントレースを選択して転送します。

そしてナビ信号を発振し，機体を線からたとえば15 c m程度離した位置に置き，RUNボタンを押すと，15cmの距離を維持しながら周回します。

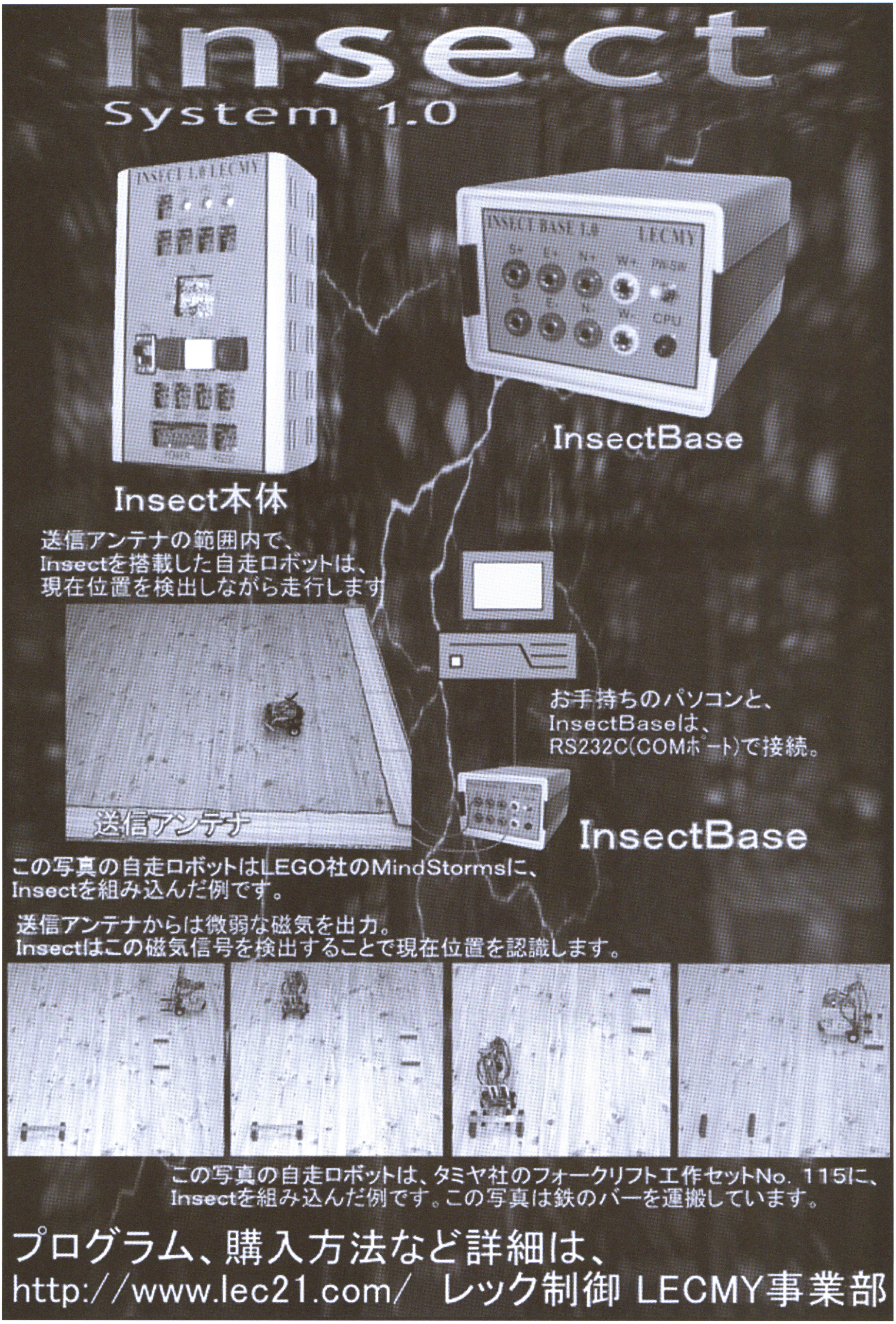
周回する方向は，円の内と外で異なります。

ライントレースは円に限定するものではありません。

第6図のような曲線であってもトレースします。

次号では，ハードウエア構成とプログラムについて詳細に解説します。

■「INSECT」は，レック制御（〒572-0802 大阪府寝屋川市打上242 中1-B-2，TEL/FAX 072-824-8608）から，一式68,000円で発売されています。詳細はホームページ http://www.lec21.com/ を参照してください。


Insect
System 1.0
InsectBase
Insect本体
送信アンテナの範囲内で、
Insectを搭載した自走ロボットは、
現在位置を検出しながら走行します
送信アンテナ
お手持ちのパソコンと、
InsectBaseは、
RS232C(COMポート)で接続。
InsectBase
この写真の自走ロボットはLEGO社のMindStormsに、
Insectを組み込んだ例です。
送信アンテナからは微弱な磁気を出力。
Insectはこの磁気信号を検出することで現在位置を認識します。
この写真の自走ロボットは、タミヤ社のフォークリフト工作セットNo. 115に、
Insectを組み込んだ例です。この写真は鉄のバーを運搬しています。
プログラム、購入方法など詳細は、
http://www.lec21.com/ レック制御 LECMY事業部

21世紀の遊びはコレ！ロボ道場

「全国高専ロボコン」レポート＆階段を降りるASIMO／二足歩行ロボット格闘競技大会情報

取材／文
松原拓也

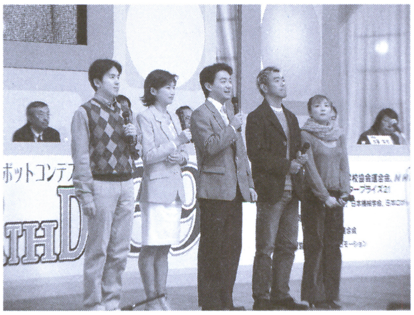
アイデア対決
全国高等専門学校ロボットコンテスト2001

主催：高等専門学校協会連合会，NHK，NHKエンタープライズ21

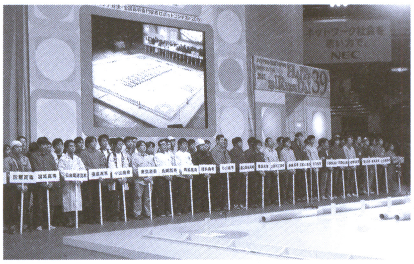
■Robocon Official Site
http://www.nep21.co.jp/robocon

「全国高等専門学校ロボットコンテスト（高専ロボコン）」は毎年，そのようすがテレビ放映されるほどの人気ロボット競技です（**写真1**）。

14年目を迎えた「高専ロボコン2001」では，全国62高専124チーム（1校あたり2チーム）が参加し，その中から24高専が選抜され，全国大会に挑みました（**写真2**）。なお，全国大会は東京・両国国技館で行われ，その模様は2001年12月21日にNHK総合TVで放送されました。



▲《写真1》有名TV番組だけあって出演者や演出も豪華



▲《写真2》選び抜かれた24校が全国大会に出場

■今年の競技課題「Happy Birthday 39」

競技は赤・青の2チームで行います。各陣地にはそれぞれ「ケーキゾーン」「ショートケーキゾーン」という場所があり，そこに「ろうそく」と呼ぶ紙製の筒をより多く立てたほうが勝ちになります（**写真3**）。

ろうそくは，ただ立てればいいというものではなく，積み重ねたり，上に「炎」というモチーフを乗せることで，得点が倍増します（**写真4**）。そのため，各チームのロボットは効率良く「ろうそく」を積み重ねられるよう，さまざまな工夫がされています（**写真5**）。

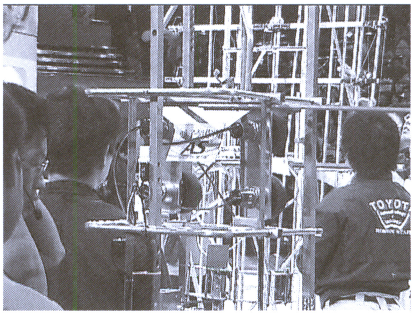
▲《写真3》競技フィールド。赤と青に陣地（ケーキゾーンとショートケーキゾーン）が塗り分けられている

▲《写真4》「ろうそく」を自陣に立てれば得点アップ。さらに積み上げたり，「炎」を乗せると得点が倍増

■マシンの規定

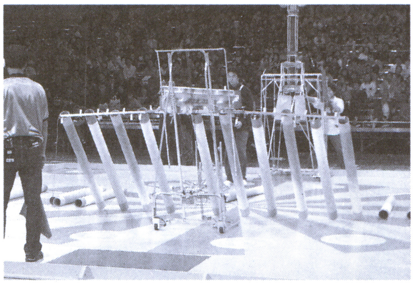
マシン（ロボット）作りには，大まかに以下のようなルールがあります。

● 今大会からマシンの操縦はワイヤレスになりました（赤外線・可視光・音波。自動制御も可）。これで，ケーブルが

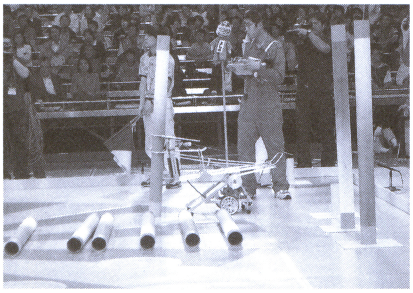


▲《写真5》ろうそくの積み重ね機構。タイヤを使って押し上げるタイプが最も多かった

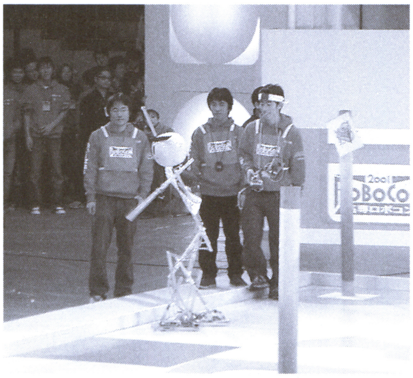
▲《写真6》高さ制限がないため，巨大なマシンが続々登場



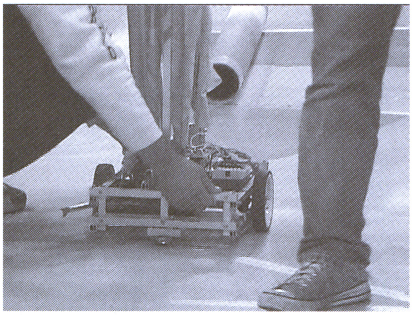
▲《写真7》一度に大量得点を狙う仙台電波高専「transfix」の串刺し機



▲《写真8》操作性が抜群だった北九州高専「すごいぞぉ」の子機

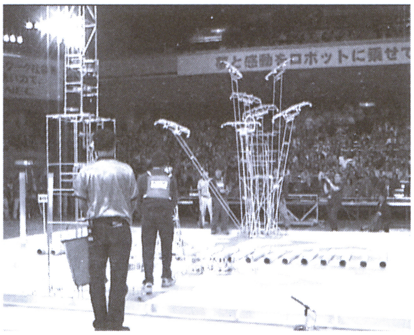


▲《写真9》二足歩行の子機が大ウケだった佐世保高専「Moon Walker」（アイデア倒れ賞）



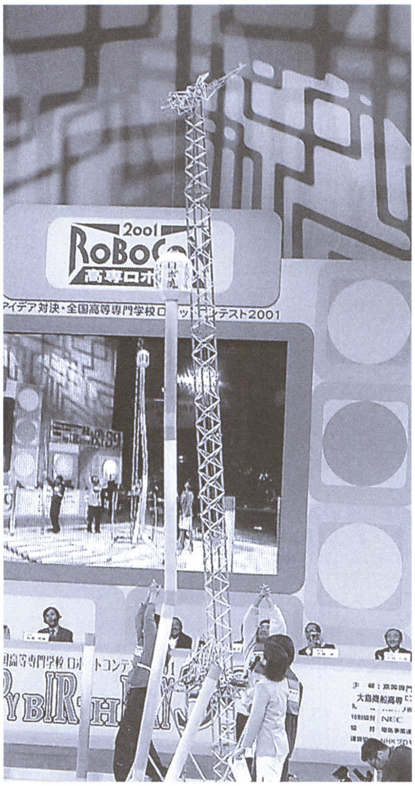
▲《写真10》一部にRCXを使っていた松江高専「技あり一本」

▲《写真11》4段重ねで72点という高得点を叩き出した仙台電波高専「transfix」（準優勝）



▲《写真12》倒れ込んで，一斉にろうそくをつかむというアイデアがすばらしい豊田高専「点火無敵」（特別賞）

▲《写真13》相手が取りにくくなるように，ろうそくの並びをくずす詫間電波高専「チャレンジャー」。なんと2年連続で優勝



▲《写真14》ロボコン大賞を受賞した大島商船高専の「オロチ3兄弟」。糸でろうそくを吊り上げるアイデアが評価された

▲《写真15》受賞に大喜びする大島商船高専の生徒さん

からまるという問題はなくなりました。

- マシンはスタートゾーン（1200×1200mm）の大きさに収まるようにします。逆に高さ制限はありませんので，いくらでも上に伸ばせます（**写真6**）。しかし，重量は27kg以下と制限されていますから，軽くて丈夫なロボットを作らないといけません。
- 構成部品の合計金額が8万円を超えてはいけません。お金をかけないように知恵を絞って作らないといけません。

■結果

大会には，巨大で大量得点を狙ったマシンや，小型で堅実なマシンなど多種多様なマシンが数多く登場しました（**写真7〜12**）。

多くの熱戦の末，香川県の詫間電波高専が2年連続優勝しました（**写真13**：ロボット名「チャレンジャー」）。

また，ロボコン大賞（アイデア・技術力の最優秀賞）では，「オロチ3兄弟」を作った山口県の大島商船高専が受賞しました（**写真14，15**）。

高専ロボコンは地元応援団の声援や，授賞式で感極まって涙を見せる学生さんなど，まるで「ロボット界の甲子園」のような盛り上がりを見せました。

なお，高専ロボコンは試合観戦者を毎年募集しています。両国国技館で実際にこの感動を味わってみてはいかがでしょうか。

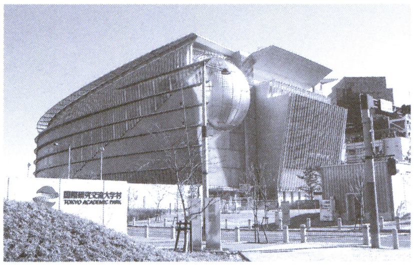
2002年は「二足歩行ロボット・ブーム？」

「夢のまた夢」と思われていた二足歩行ロボットですが，今年はさらにそのブームがやってきそうな勢いです。ここでは，二足歩行競技大会のお知らせと，「ロボフェスタ神奈川2001」で見つけた二足歩行ロボットを紹介しましょう。

世界初!! 二足歩行ロボット格闘競技大会「ROBO-ONE」 2002年2月2～3日開催!!

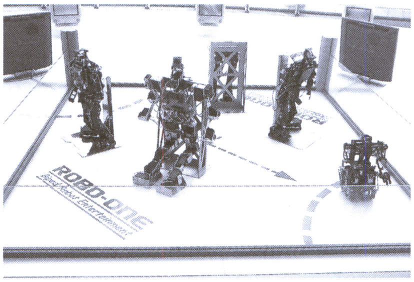
二足歩行ロボットによる競技大会「ROBO-ONE（ロボワン）」が2002年2月にはじまります。競技スタイルはなんと格闘技。リングの上で2体のロボットがパンチやキックなどを繰り出して戦います。

32台の参加ロボットの製作者はみんな趣味として物作りをしている人達（ホビーイスト）です。ROBO-ONEは参加して楽しい，見て楽しい，ロボット作りの楽しさを伝えるような競技会を目指しているそうです。



▲会場は日本科学未来館

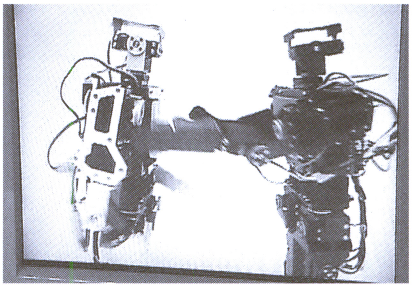
開催日：2002年2月2日(土)～3日(日)
会　場：日本科学未来館1階シンボルゾーン
主　催：日本科学未来館
運　営：ROBO-ONE委員会
事務局：株式会社ベストテクノロジー
入場料：日本科学未来館入館料に含まれる(大人500円/18歳以下200円)



▲ROBO-ONE用のリングと自作のロボット達（大会当日には32体のロボットが集まる予定！）

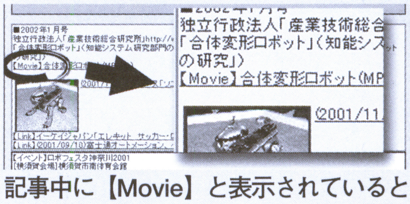
◆ROBO-ONE公式ページ
http://www.robo-one.com/
◆日本科学未来館
http://www.miraikan.jst.go.jp/

▲ベストテクノロジーの五味渕弘毅さん。手にしているのは第1回ROBO-ONEのシンボルロボット「FREEDOM」。一般発売もしており，一体48万円（受注生産）

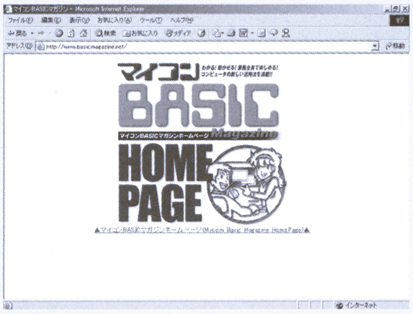


▲試合のイメージ映像。ロボットが1対1で格闘

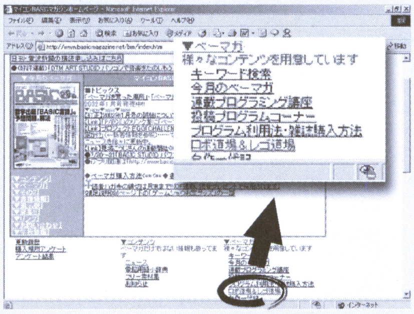
ここで紹介したロボットの動画を本誌ホームページからダウンロード！



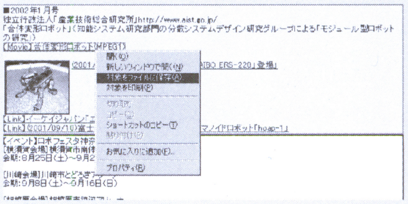
記事中に【Movie】と表示されているところにマウスを持ってきて，右ボタンをクリック（写真では1月号の映像ファイルのダウンロード部分を示していますが，【Movie】マークが目印です）



http://www.basicmagazine.netにアクセス！



「ベーマガ」の「ロボ道場＆レゴ道場」をクリック

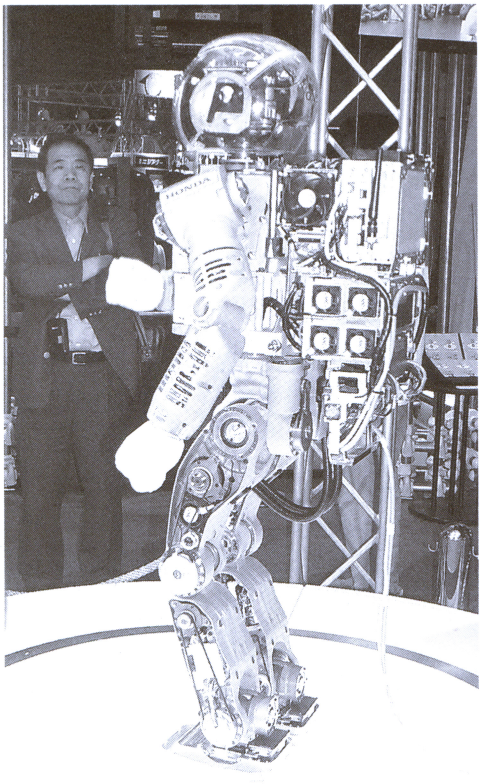


ウインドウが開き，その中の「対象をファイルに保存」をクリック

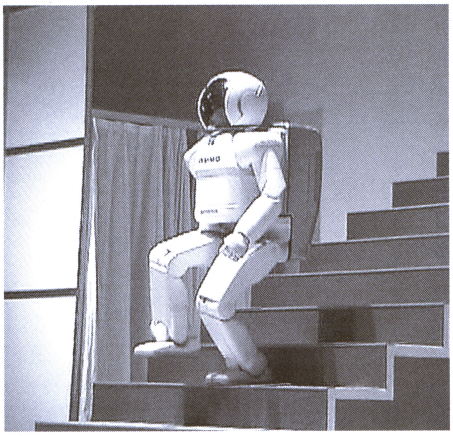
ロボフェスタ神奈川2001 横浜会場
階段を降りるASIMOが登場

先日行われたロボフェスタ神奈川2001横浜会場では，多くの二足歩行ロボットが展示されていました。

2月に開催されるROBO-ONEや，2002年6月に行われるRoboCupのヒューマノイドリーグなど，今後の二足歩行ロボットの勢いは続きそうです。

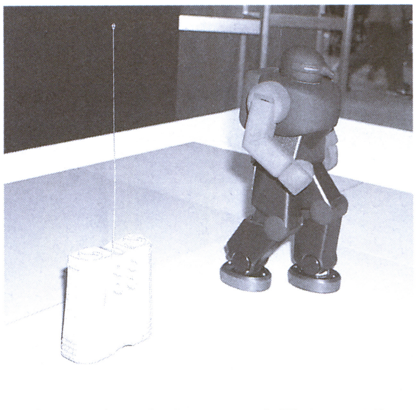


▲ベルト，モータ，冷却ファンがむき出しのホンダ「ASIMO」

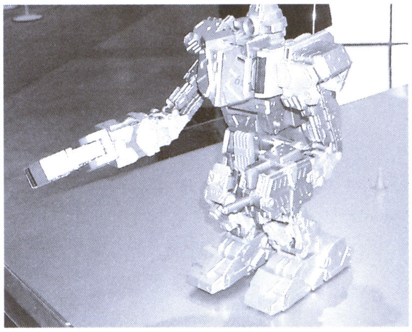


▲階段の昇り降りができるようにバージョンアップしたホンダ「ASIMO」。25段もの階段を降りるデモは圧巻！

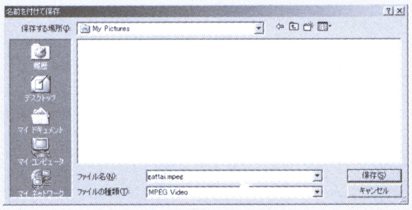
▲BB弾発射やカメラ映像をプロポに投影できるラジコンザク。バンダイ「RC TECH ROID MS-06F ZAKU Ⅱ」。価格は98,000円（税別）。予約受付中。2002年2月発送予定



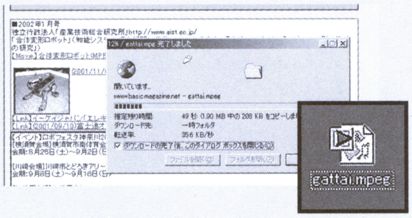
▲京都大学の学生である高橋さんが作った「MAGDAN」という二足歩行ラジコンロボット。この技術が京商の「ガンウォーカー」という商品で使われる予定



▲BB弾発射やクランプハンド，ダメージセンサや双方向通信機能を搭載したラジコンロボット。タカラ「ドリームフォース01」。価格は49,800円（税別）。発売中

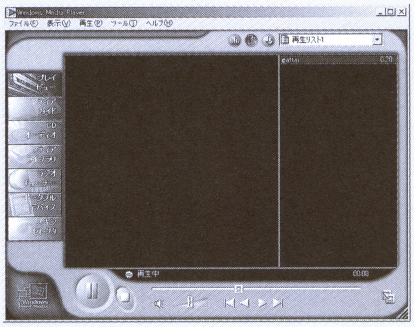


保存する場所を選んで，保存ボタンをクリックすると，



ダウンロードが始まり，終了すると，アイコンが現れます

これを「RealPlayer」か「Windows Media Player」で再生してください。

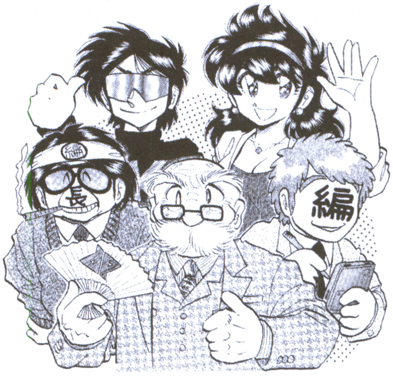


▲Windows Media Player

※【Movie】マークをそのままクリックした場合，ダウンロードが自動的にはじまり，設定されている「RealPlayer」また「Windows Media Player」などが自動的に立ち上がります。

▲Real Player

あくせすROOM



新入生勧誘大作戦

影：もうすぐ春。新入生がいっぱい。電波新聞社も，夢と希望に満ちた新入社員が入ってくる季節だぞ！とりあえずパソ破クラブで新入りを何人か確保しようと思うのだ。

編：そんな社内サークルは聞いてませんよ。ええと，……パソハ？　パソハ，パソハ。

影：一子相伝の影拳をキワめた私には，指先一つであらゆるパソコンをダウンさせられるという，呪われた能力が秘められているのだ。免許皆伝を記念して，この技「パソコン破壊（パソ破）」をこれからパソコンを仕事でフル活用する新入社員に伝授して，給料を倍にしないとパソコンをダメにするぞーとか労働交渉の材料にするのだ。

編：それは単に脅迫と言うのでは……。

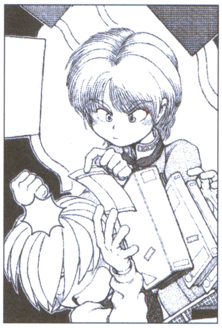
影：しかし，少子化で学校も生徒数が減っている。サークルの主催者としては，新人の勧誘にいろいろ策を練らねばならないぞ。

編：たしかにそうですね。気持ちはわかります。

影：そして，我々にはパソコンがある！（持ってない人も多いけど）　それを使って，デモを展示したり，チラシを作ったり……いろいろできそうじゃないか？　いや，できる。読者諸氏にいままでの実績，今年の計画などを送ってもらって，参考にしあうというのはどうだろう。

編：ああ，それでは，4月号（3月8日発売）の募集テーマはそれにしましょうか。締め切りは，2月15日ぐらいでいいですか？

影：しかし，最近は投稿少ないね……。みんな受験とか期末テストで忙しかったのかな？

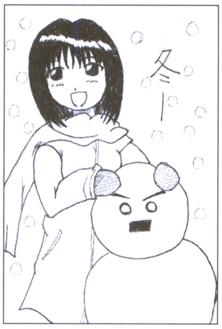


▶東京都練馬区・どらねく

影：雪だるまを作るゲームとかも，おもしろいかも……。

編：お，ひさびさのブラック・ボックスですか？

つぐ美：寒い冬があったかく感じられるイラストです♡　あとは下書きさえちゃんと消してくれていれば……。



◀愛知県刈谷市・わいち301

編：こづかれた衝撃でさまざまな思いが走馬燈のごとく……というわけではありませんが，お年玉の背景に凧が大量に舞っているのがシュールです。正月にお年玉でゲームソフトを買い込んで，1年持たせるってのはすごい！

影：あのころはソフトも少なかったし，新製品には必ずなにか今までにない新しい要素とか，インパクトがあったよね。

つぐ美：あのころっていつよ。

◀岐阜県可児市・サボタン

編：集会に公式も非公式も関係ないですよ！

つぐ美：初の読者集会参加。ドキドキだった？　みんなはしゃぎすぎて，ケガとかしないでね。参加した人のレポートお待ちしていま～す。

▲大阪府吹田市・ARAI

影：古代ローマの闘技場で戦っていれば，演出として虎が使われるかも。

編：映画「グラディエーター」とかですか？　あのトラはロボットですよ。たくさんいるローマ兵士も，ほとんどがCGで増やされています。ローマ市街の再現や，戦場の炎などでもコンピュータが大活躍ですね！

あくせすROOM応募のルール

●募集物

【時期限定】

3月号（2月8日発売）

- うろおぼえ友の会「パールのようなもの」。この謎のアイテムについて，想像をめぐらせたイラストなどを募集（2002年1月15日必着）

4月号（3月8日発売）

- うろおぼえ友の会「トウヘンボク」「ボクネンジン」どちらかで結構（2002年2月15日必着）
- テーマ企画「新入生勧誘大作戦」（2002年2月15日必着）

【常時募集】

- パソコンの都市伝説（パソコンにかかわる信じられない事件や，ウワサ話）
- 学校であったできごと
- 投稿作品に関する感想・意見
- サークルメンバーの募集などなど

●あくせすROOMでは，みなさんの自由な題材での投稿をお待ちしています。

ワンポイント

ゲームの話題，学校の話，プログラムの小ネタなどなど，テーマは自由。最近は文章投稿が少ないので，とくに狙い目のようです。

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

ワンポイント

まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

ワンポイント

下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫

〒141-8715 東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

「第4回CESAアマチュアゲーム大賞～めざせ！ 未来のクリエイター」開催決定

社団法人コンピュータエンターテインメントソフトウェア協会（略称：CESA）が，アマチュアゲームクリエイターが制作したオリジナル作品の中から優秀作品を選考する「第4回CESAアマチュアゲーム大賞」の開催を決定。同賞は，非商業ベースでゲームを作っているクリエイターの作品を表彰することで，コンピュータエンターテインメントソフトウェア産業の未来の人材を発掘・育成することを目的としている。

＜募集概要＞

募集期間：2002年3月1日（金）～2002年6月末日 ※必着

応募資格：オリジナルゲームを制作しているアマチュアの個人またはグループ

対象作品：オリジナルで市販されていない作品（原則としてプラットフォームの限定はないが，携帯電話のゲームは，作品に加えて実機を送ることで，審査で実演できることが応募条件）

審査員：プロのゲームクリエイター，ゲーム専門雑誌編集者

表　彰：●大　賞　1作品
●優秀賞　1作品
●部門賞　3作品

発　表：「東京ゲームショウ2002秋」内イベントステージ（予定）

※詳細は，公式サイト「http://www.cesa.or.jp/」を参照。

問い合わせ先：
CESA事務局
〒105-0003　東京都港区西新橋1-22-10西新橋アネックス3階
電話　03-3591-9151
FAX　03-3591-9152
URL　http://www.cesa.or.jp
e-mail　info@cesa.or.jp

2月号 お楽しみプレゼント

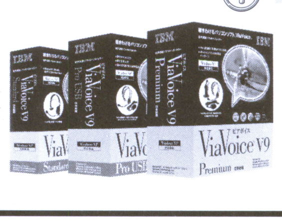
①「ビアボイスforWindows」
（話した内容をワープロ・ソフトなどにそのまま入力できる音声認識システム搭載ソフト。2001年11月22日にバージョン9が発売された）
A Standard版＝Wordの文章入力中心に考えている人向け
B Pro USB版＝音声コマンド入力でパソコン操作も快適に活用可能
C Premium版＝Pro USB版にノイズ償却機能付き手持ちマイク同梱
各1本3名（提供：日本アイ・ビー・エム）
※①—A，B，Cいずれか選んでください。
②プライズ用グッズ各種／5名（提供：編集部）
③電波新聞社オリジナル・カレンダー／5名（提供：編集部）

●応募方法：本誌とじ込み愛読者ハガキ（またはインターネット読者ハガキ）にプレゼント記入欄があります。一つだけ選んでお送りください。

●締め切り：2002年2月7日必着

《2001年12月号の当選者》①新潟県新発田市・菱沼竹宏さん他9名 ②北海道札幌市・早苗春菜さん他1名 ③東京都江戸川区・千葉雄太さん他1名

①

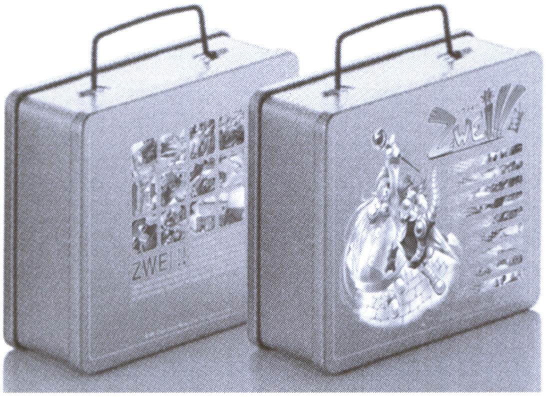


Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714
スタッフ募集中
採用担当BFN係
http://www.falcom.co.jp/

「Zwei!!（ツヴァイ）」五つめの特典！？ブリキ缶CD-BOXプレゼント！

■毎月抽選で合計800名にプレゼント！■



—初回特典版は製造終了！店頭在庫のみですのでお早めに！—

遂に発売された「Zwei!!（ツヴァイ）」ですが，もうクリアされましたか？Zwei!!はエンディングを見たらおしまいではありません。2周目以降にしか登場しない敵やアイテムもありますし，図鑑のコンプリート，ミニゲームなど，いろんな楽しみ方があります。最近の海外ゲームに代表されるダークな雰囲気ではなく，親しみやすい明るく元気なRPG「Zwei!!」を，まだプレイされていない方は，ぜひ遊んでみてくださいね！

今回は，初回特典として，同時発売されたFMS第5弾の，Zwei!!の全曲が収録された「オリジナルサントラZwei!!」（2枚組）と，完全新作新アレンジの"癒し系"「イースヒーリング」の2タイトルに加え，パッケージは特製シルバーメタリックBOX，そしてピクチャーレーベルと，4大特典が付いていますが，さらに，それだけではなく，購入された方でアンケートハガキを送ってくれた方に，毎月抽選で合計800名に，Zwei!!の特製ブリキ缶CD-BOXをプレゼントします。ケースのまま5枚のCDが収録でき，ドライブや旅行に行く時にCDを持ち運ぶのにすっごく便利なアイテムです。

詳しくは，パッケージに同梱されているアンケートハガキをご覧ください。

試してZwei!! キャンペーン実施中！

デモは見たけど，実際操作してみないとわからないという方のために，体験版を用意しました。パソコン誌の付録CD-ROMに収録されてありますし，Webサイトでもダウンロードできる予定ですので，ぜひZwei!!を実際にプレイして体験してみてね！

体験版収録誌

●DOS/V POWER REPORT（12/27売）●TECH Win（1/8売）●CD-ROM Fan（1/8売）●コンプティーク（1/10売）●ログイン（1/24売）

Zwei!!の着メロ 好評配信中！

発売と同時に，ファミ通comで配信が開始されたZwei!!の着メロが，早くも大人気！今回配信されたのは下記7曲。

「永劫の夢，大空の記憶－Zwei!!2人で大冒険－」
「ブック村」
「浮遊大陸アルジェス－Main Theme－」
「ひとときの休息を」
「おやすみ」
「妖精たちの村」
「花と風のうた」

Zwei!!はもちろん，これからもどんどんファルコムサウンドが着メロで登場しますのでお楽しみに！

詳しくは下記サイトをご覧ください。
http://www.famitsu.com/mobile/melomelo/

イラスト集，好評発売中！

ファルコム初のイラスト集「Falcom History Legend of Illustlations」が，先月発売。イラストはファルコムファンにはお馴染みの，末弥純，米田仁士，西村博行，寺田克也（敬称略）他の，豪華な作家陣だけでなく，フィギュアや，CG作品，未発表のイラストも含めたファルコムの歴史が綴られた永久保存版のイラスト集です。

全国どこの書店でも購入できますので，店頭にない場合はお店の人に取り寄せてもらおう。ファルコム通販でも購入できますのでぜひご利用ください。

発行：ビレッジセンター出版局
http://www.villagecenter.co.jp/book/art1-falcom.html
価格：2800円（税別）
Code番号：ISBN4-89436-140-X

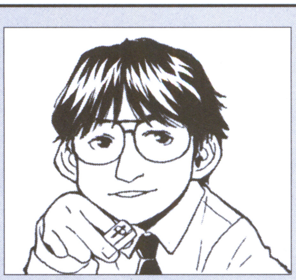
2002年のファルコムは…？

現在，「イース」「英雄伝説」といわばファルコムの顔とも言える超大作が開発進行中。英雄伝説は，海の檻歌でガガーブトリロジーも完結したので，全く新しい世界の，新キャラクターの物語となります。「イース」に関しては，開発のガードが固いので，まだ詳しい話ができませんが，かなり期待できそうです。

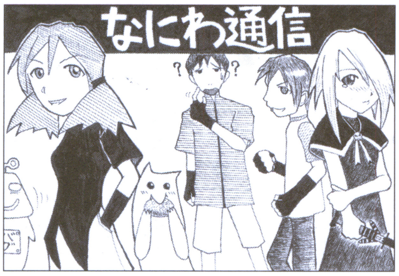
さらに，ネットワーク対戦のあのゲームの新作や，移植希望No.1の3DダンジョンRPGがポリゴンMAPとなっての登場，さらに極秘プロジェクトも進行中と，今年は，個人的にも仕事を忘れてプレイしたいタイトルが目白押しとなっていますので乞うご期待！

突然新作の情報が公開されたりするので，ファルコムホームページは常にチェックしておこう！
http://www.falcom.com/

▲これが開発中の新作ゲーム画面だ！カラーじゃないのが残念。



1/25日は，Zwei!!の攻略本が発売されますので，完全制覇を目指している方は，ぜひご参考に。CD-BOXもサンプルを見せてもらいましたが，かなりいい感じに仕上がっていますので，ぜひGetしてね！
今年もファルコムの応援をよろしくお願いします。



タイトル冠文：大阪府・御堂筋太郎
タイトル・イラスト：千葉県・バグニンゲン
本文：暁

初詣の後は なにわ通信

PRESENTED BY マイコンソフト
http://www.micomsoft.co.jp/

ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F 電波新聞社大阪本社内
マイコンソフト株式会社 なにわ通信係

明けまして，おめでとうございます！2002年になってしまいましたね〜。

本年も，なにわ通信を宜しくお願いいたします。

ごめんなさい

新年早々，皆さんには謝らなければいけません。

昨年末，「年越しそばを食べ終わるまでには必ず発売する」と申しておりました，新製品の「XRGB-2plus」ですが，残念ながら，12月中には発売することができませんでした。1月下旬には必ず！必ずや発売させていただきます。誠に申し訳ありません。

「XRGB-2」のこんな使いかた

「XRGB-2plus」はまだ発売されておりませんので，旧型機「XRGB-2」の面白い使いかたをご紹介しましょう。XRGB-2にも，まだまだ使い道が残されています。ビデオ・キャプチャ機能を持つパソコンを使用されている方には，うれしい話題です。

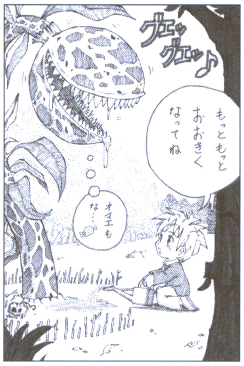
現在販売されている多くのビデオ・キャプチャカードやビデオ・キャプチャ機能付きのパソコンには，入力された映像に対してガンマ補正を施すような機能がありません。

このため，キャプチャ・ソフト等で画面の明るさを調整すると，画面の明るい部分が真っ白になって色が飛んでしまったり，画面を暗くすると，薄暗い部分が真っ黒になってしまい，思うように調整できません。

実際に，私もキャプチャ機能を使用して，パソコンでテレビを見ていますが，納得のいく画質が得られたことはありません。

もし，同じような悩みを抱えていて，すでに弊社の「XRGB-2」と「XAV-2s」をお持ちの方がおられましたら，下図のように接続して，ゲーム機などの映像にガンマ補正をかけてみてください。少しは見やすい画面でキャプチャすることができますよ。

4コマ漫画も募集中のありがたいおハガキのコーナー

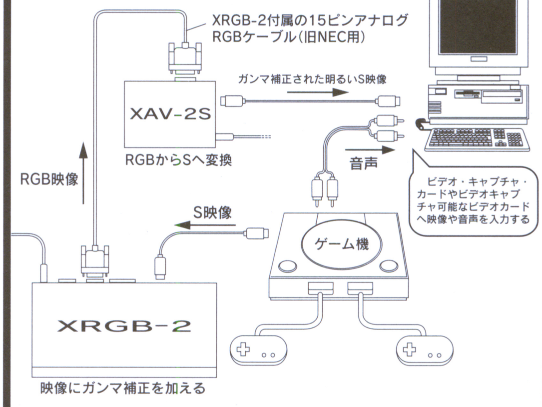


千葉県・鈴木有人
前・後編の1コマ漫画。今回は前編。この怪しい植物は坊やのペットらしいです。それにしても表現方法がうまいですねぇ。

福島県・葵鉤菜
可愛いらしいイラストですね。このままゲームの宣伝に使えそうなデザインです。元ネタはゲームですか？

山形県・まよ十こぐま
赤ちゃんご誕生おめでとうございます。こぐまさんに宜しくお伝えください。こぐまさんも，こんな絵が描けるようになるのでしょうか。

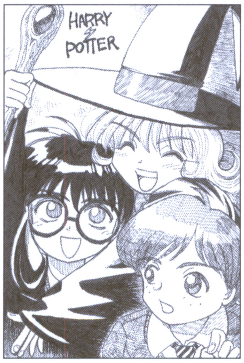
ゲーム画面をガンマ補正して明るくキャプチャする方法



ご注意：XRGB-2の出力モードは15kHzに設定します。
スクリーン・メニューの「GAMMA」機能は，「3」や「4」に設定しておきます。

埼玉県・ぶんちょ
ちょっとブルーな朝でしょうか。それとも，低血圧？朝の日差しがうまく表現できていると思います。

大阪府・ARAI
ついに最終話です！なんと悲しげなイラストでしょうか。みんな海へ消えてしまった…。
漫画公開希望！。



埼玉県・野沢プロト
ハリーポッター，私も見に行きたいですっ！魔法学校の入学金ってどれ位必要なんでしょうか？
魂1個で足りますか？

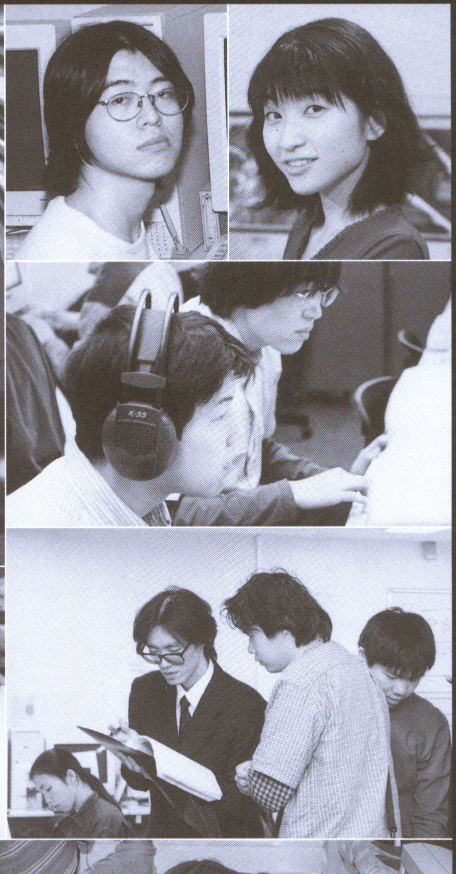
京都府・陽東あひる
昨年の獅子座流星群は見ましたか？残念ながら，私はぐっすり寝ていました。
腰の調子はどうですか？

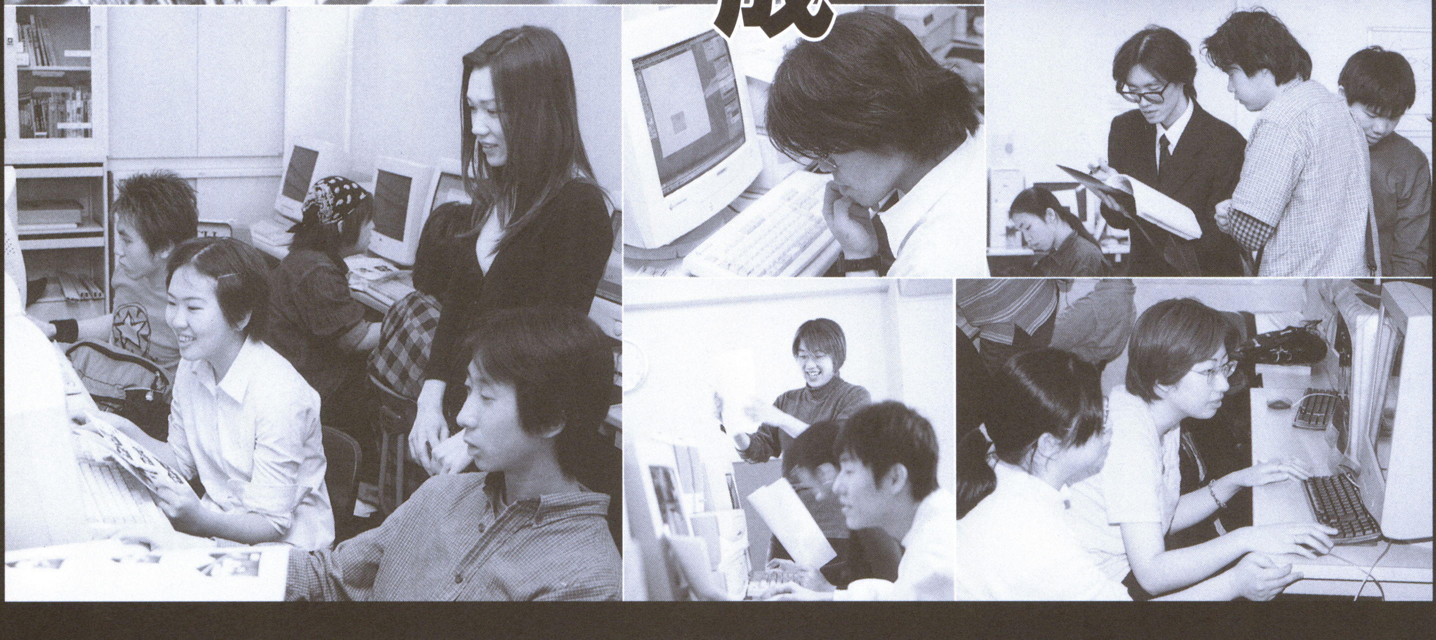
徳島県・るじパルコ
今までイラストありがとうございました。ベーマガがあなたのお役に立てて良かったです。どうかいつまでもお元気で。

今年こそ目標達成

★ここが違う！現役ゲームデザイナーが、就職できる技術を徹底指導!!

●ゲームデザイナーになる！厳しくても好きなことだからがんばれる!!





授業内容
各科共に年間カリキュラムが組まれ、それを前後期で実施。2年生はチームを組んで現場と同じにゲームを制作。ここで就職できる実力を習得している。

講師
本学院の講師はすべて現役のゲームデザイナーが指導。業界に精通しているので、就職情報や技術面でも生徒の信頼が厚い。現場への研修も斡旋する。

マシン環境
ゲーム業界はソフト、ハードとも常に変化している。学院では毎年、最新設備を整え新入生を迎えている。現場で使われている機材で学習するのがベスト。

就職活動
現在、3月卒業にむけて就職活動もピークをむかえている。生徒が希望する就職を達成させるために、進路指導部と担任の先生が、個別にきめ細かな指導をしている。幸い求人数も多く、生徒の技術力によって、それなりの就職内定を獲得している。（13年12月現在）

学費（年間費用）
創立以来、ゲーム関連学校の中では、常に良心的で適正低廉な学費を設定。14年度入学生は84万円、2年目は66万円。入学後の徴収はなく、すべてを含む。

教育ローン・学生寮
①本学院が三井住友銀行と提携している教育ローンだと、毎月6276円で学べる。80万円を14年間返済、②環境と設備を整えた学生寮（男女各90名）あり。

新聞奨学制度
朝日、毎日、読売新聞等を配達することで、学費が全額免除される。また、住む場所と給与12万円が支給される。手続きなど、詳しくは入学案内書に記載。

体験実習・学院見学
体験実習日は1月12日・26日、2月9日（いずれも土曜日）午後1時開催。参加希望者は電話でお申込み下さい。学院見学は毎日実施。電話で要予約。

■就職抜群!学生寮完備。朝日・毎日・読売新聞等奨学金制度（学費全額免除・給与月額12万円支給）三井住友銀行教育ローン有。学院見学入学相談随時可⇨日時をTELにて要予約

14年4月生願書受付中
ゲームデザイナー学部
全日制2年・定員300名

ゲームデザイン科
プログラム科
グラフィック科
キャラクターデザイン科
デジタルサウンド科

文部省認可CGARTS協会認定CG教育校／社団法人・日本漫画家協会賛助会員／社団法人・日本映画テレビ技術協会会員

日本ゲームソフト学院

〒101-0061 東京都千代田区三崎町3-4-9 ☎03-3261-8211

入学案内書 はがき（住所・氏名・年令・電話番号明記）か、電話で上記まで（無料）

FAX 0120-194-750 http://www.tag.co.jp/ E-mail:ngs@tag.co.jp

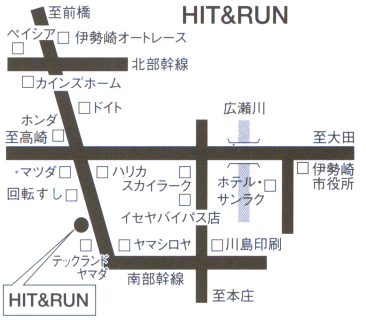
北関東地区DO

「マイコンBASICマガジン」おすすめショップ

北関東のDOS/Vショップは今日も元気に頑張っています！
女性店長あり、話好きの店主ありと個性豊かな店ばかり。
変わらないのはみんなパソコンが好きだということ。
一度、近くのショップに立ち寄ってみては。
親切丁寧に応対してくれますよ。

① HIT＆RUN

http://www.leoplanet.co.jp/hit/



ショップ営業時間
AM10:00〜PM7:30（火曜定休）

- ●お問い合わせ、ご注文はインターネット、又はTEL、FAXにても受付けております。
- ●通信販売での支払法
  銀行振込、ヤマト便の代金引換払いにてお願いします。
- ●本広告を見てご来店、購入の方にPS/2マウスプレゼントします。

〒372-0813 群馬県伊勢崎市韮塚町561-1
☎0270-21-1491　FAX 0270-26-9561

取扱い品

- ●オリジナルPC組立キット（各種）
- ●オーダーメードPC組立キット
- ●組立費無料（但し、￥50,000以上セット）、OSインストール費￥2,000より
- ●組立式ノートパソコン
- ●各種DOS/Vパーツ全般

メールマガジンサービス

“PC&PARTS価格ウォッチ”
毎週の調査により、週1回配信
現在読者数13,000名
http://www.leoplanet.co.jp/pc/

中古部品 高値買取り

- ●各社マザーボード
- ●各社ビデオカード他
- ●各種ドライブ類他
- ●ノートPC，周辺機器

ノートPC HDDグレードアップサービス

HDD交換サービス（メーカー保証対象外）
- ●10GB HDD＋取付＋データ転送￥22,600
- ●20GB HDD＋取付＋データ転送￥27,900

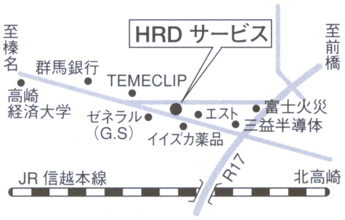
業者在庫処分メーリングリスト開催

無料会員、業者会員募集中!!
http://www.circle.ne.jp/ml/outlet/

自作パソコン＆パーツのお店 HRDサービス

秋葉原価格に挑戦

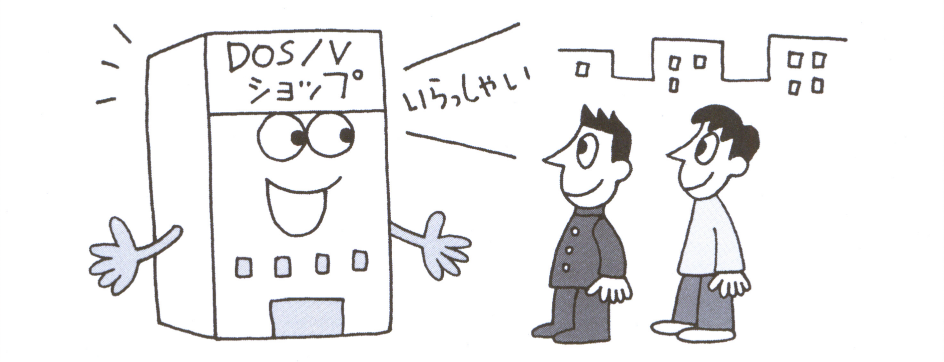
インターネットで最新情報をGET!
http://www.hrdservice.com



- ●メンバーズカード会員募集中!!会員特価でお得!!
- ●中古ジャンクションコーナー新設
- ●パソコン出張サービス
- ●月額3000〜のホームページ開設サービス（初回手数料別）
- ●最新情報をタイムリーに提供
- ●サーバ構築にも多数の実績
- ●CD-Rメディアは当店へ

〒370-0801 群馬県高崎市上並榎町479-4 ☎027-370-3700 ②

S/Vショップマップ

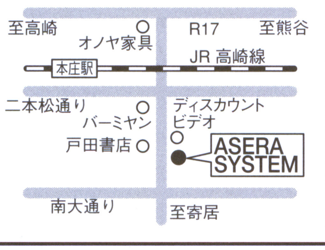


群馬県勢多郡富士見村大字小暮2225-19
TEL 027-288-9473
http://www.garagecom.jp/
mailto:masac@bh.mbn.or.jp

3

ガレージコム

ASERA SYSTEM



埼玉の田舎でガンバってます！
パーツ・組立PCのことなら当店へ

DOS/Vアセラシステム

ASCAL

埼玉県本庄市北堀866-1
Tel/Fax 0495-23-0379（火曜定休）
http://www.ksky.ne.jp/~asera/

4

シグナル

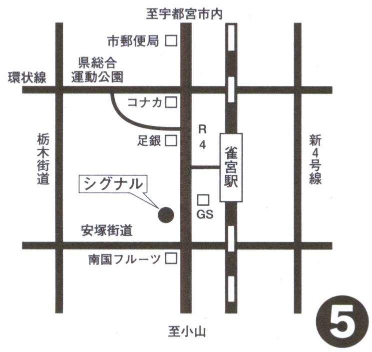
http://www.signal-pc.com

DOS/Vパソコンはじめての方、お店で組み立て、お教えいたします。

最新情報はホームページで！

営業時間
10：00〜19：00（日曜休、祭日営業）

〒321-0132
宇都宮市雀の宮4-5-12
☎ 028-655-2983
FAX 028-655-2969
e-mail:signal@signal-pc.com



5

中古通販 PC NET ピーシーネット

在庫保有数日本最大級（常時10,000台）

TEL.03-3733-1441

通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30

www.prins.co.jp/

楽天市場店 http://www.rakuten.co.jp/pcnet/
ネットプライスモール http://www.netpricemall.com/
ジャストネット http://email.justnet.ne.jp/shop/pcnet/
ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

最新情報は電話かネットでチェック!! 急げ!!

プロジェクター大放出!!

PC NET イチオシ商品!! 在庫数あとわずか!!

エプソン U8474
ELP-5000
解像度:800×600/明るさ:500ANSIルーメン/投影サイズ:23～300インチ/液晶パネル:1.3インチ×3/外寸:245×380×159/重量:6.4kg/
発売時期:96/11 ¥898,000
¥128,000

エプソン U5216
ELP-3500
解像度:640×480/明るさ:650ANSIルーメン/投影サイズ:21～300インチ/液晶パネル:1.3インチ×3/外寸:283×382×137/重量:6.7kg/
発売時期:96/11 ¥698,000
¥99,800

富士ゼロックス U5225
XP600S
解像度:800×600/明るさ:600ANSIルーメン/投影サイズ:30～300インチ/液晶パネル:1.3インチ×3/外寸:421×300×163/重量:9.8kg/
発売時期:97/06 ¥898,000
¥128,000

SANYO U8475
LP-SG50
解像度:832×624/明るさ:550ANSIルーメン/投影サイズ:20～400インチ/液晶パネル:1.3インチ×3/外寸:296×460×149/重量:7.2kg/
発売時期:96/10 ¥848,000
¥99,800

インターネットでは毎回お得な企画がいっぱい!!

3日間保証セール だから安い!!

期間限定タイムセール

全商品 外装クリーニング 無料! サービス

[安心の全品保証付] 1万円以上…1ヵ月保証 1万円以下…1週間保証

※本商品は添付品（OSを含む）は付属しておりません。
※中古パソコン販売時の在庫定義が変わりました。

今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充（他メーカー含）されている商品も、「在庫」としてお取り扱いさせていただきます。

DOS/V

注文No.	商品	価格
U8905	Celebris GL-2(DEC)(PenII(266)/32MB/3.2GHD/CD16/外装焼け/スロットカバー無し/汚れ/ミドルタワー型)	¥13,800
U8932	PC5000(PC5510/6300ML)(DEC)(PenII(300)/128MB/3.2GHD/CD)………	¥20,800
U8344	Aptiva(2190-27N)(IBM)(K6-2(500)/60MB/10GHD/CD/スタンド無し/外装焼け/省スペース型/PCI×3/USB×2)	¥27,800
U8525	PC-VS26DS7DA1(NEC)(PenII(266)/32MB/3.2GHD/CD24/PCI×3/USB×2)	¥16,000
SH794	PC-MA36HC5AMB43◆Win95付(NEC)(Celeron(366)/64MB/4.3GHD/CD24/省スペース型/付属品付)	¥26,800
B0182	PC-MA45DSZAMA63(NEC)(PenII(450)/32MB/6.4GHD/CD32/100TX)……	¥32,800
U8974	PC-VE500J27C,D◆Win98付(NEC)(PenIII(500)/64MB/20GHD/CD40/スピーカ無/マニュアル付/省スペース型/PCI×1/USB×2)	¥49,800
SH803	PC-MA55JMZBBBE4◆NT付(NEC)(PenIII(550)/128MB/18GHD/CD24/ミドルタワー型/付属品付)	¥53,800
U1766	PC-MA23CS(NEC)(MP(233)/64MB/3.2GHD/CD24/LAN/外装汚れ/PCI×4/USB×2)	¥13,800
U1912	PC-MA26DS(Z,7)AAA(NEC)(PenII(266)/64MB/4.3GHD/CD24/外装焼け/32MB増設/PCI×4/USB×2)	¥19,800
U8651	PC-MA40D/M-82(NEC)(PenII(400)/64MB/8.4GHD/CD32/ミドルタワー型/PCI×5/USB×2)	¥35,800
U8779	PC-MA80T/C(液晶セット)◆送料2個口(NEC)(PenIII(800)/64MB/20GHD/CD40/15TFT/KB・マウス無/省スペースセットモデル)	¥98,800
U8995	DESKPRO 4000/6266X/3200(COMPAQ)(PenII(266)/32MB/3.2GHD/CD24/外装焼け/PCI×3/ISA×3/共有×1/USB×2/AGP×1)	¥15,900
U8743	DESKPRO EP C266/3.2(COMPAQ)(Celeron(266)/64MB/3.2GHD/CD/PCカバー外装塗装剥げ/縦・横共用)	¥13,800
U8961	DESKPRO EP6350/6,4(COMPAQ)(PenII(350)/256MB/6.4GHD/CD/外装焼け/省スペース型/PCI×2/ISA×1/USB×2)	¥27,800
U1314	DESKPRO EP6466C/6.4/CDS(COMPAQ)(Celeron(466)/64MB/6.4GHD/CD32/擦り傷/ミニタワー共有/PCI×4/ISA×1/共有×1/USB×2)	¥35,800
U8807	DESKPRO C533(COMPAQ)(Celeron(500)/64MB/10GHD/CD/省スペース型)	¥36,800
U8983	DESKPRO EXD/815(COMPAQ)(Celeron(700)/256MB/20GHD/CD/省スペース型/PCI×3/USB×2/AGP×1)	¥69,800
U8977	OPTIPLEX GXA(DELL)(PenII(233)/64MB/3.2GHD/CD/外装焼け/ミドルタワー型)	¥14,800
U1135	OPTIPLEX GXA(DELL)(PenII(266)/64MB/3.2GHD/CD/高さ10.9cm)……	¥16,800
U8458	OPTIPLEX GXA(DELL)(PenII(300)/64MB/3GHD/CD24/ミドルタワー型(44.4cm)/PCI×3/ISA×2/共有×2/USB×2)	¥22,800
U8950	OPTIPLEX GXA(DELL)(PenII(333)/128MB/6.4GHD/CD/擦り傷/マウス欠品/ミドルタワー型)	¥24,800
U8907	Dimension XPS D300(DELL)(PenII(300)/64MB/3.2GHD/CD/外装焼け/ミドルタワー型)	¥19,800
U8969	Dimension XPS R350◆NT4.0付(DELL)(PenII(350)/128MB/6.4GHD/CD32/外装キズ/マニュアル付/ミドルタワー型)	¥29,800
U8941	Dimension J500C◆WinNT付(DELL)(Celeron(500)/256MB/13.6GHD/CD/省スペース型)	¥41,800
U8678	Dimension V433C(DELL)(Celeron(433)/64MB/6.4GHD/CD32/汚れ/擦り傷/ミドルタワー型)	¥29,800
U1975	GATE WAY2000(GP6-300)(GATEWAY)(PenII(300)/128MB/6.4GHD/CD24/外装焼け/タワー型(47cm)/PCI×3/ISA×1/共有×1/USB×2/AGP×1)	¥24,800
U8948	GATE WAY(GP6-450)(GATEWAY)(PenII(450)/64MB/6.4GHD/CD32/高さ48cm/マニュアル付/ミドルタワー型)	¥29,800
V0144	ZHP-D6582N/801◆Win95付(HP)(PenII(400)/128MB/6.4GHD/CD32/100BT/電源ケーブル用変換コネクター無/付属品付)	¥32,800
U8794	ENDEAVOR PRO-400S(エプソンダイレクト)(PenII(400)/32MB/2.1GHD/CD/ミドルタワー型)	¥24,800
SH811	FMV-9D10A11◆NT4.0(富士通)(K6(200)/32MB/3.2GHD/CD20/付属品付)‥	¥15,800
U1618	FMV-1DX7151(富士通)(PenII(266)/32MB/3.2GHD/CD24)……………	¥19,800
SH810	FMV-7T90B11◆NT4.0(富士通)(PenII(300)/32MB/4.3GHD/CD14/ミドルタワー型/付属品付)	¥31,800
U8745	FMV-4DXJ101,161,151(富士通)(PenIII(500)/64MB/10.2GHD/CD40/PCI×2/ISA×2/USB×2)	¥43,800
SH791	PCC-AY450N4D◆NT4.0付(パナソニック)(PenIII(450)/128MB/6.4GHD/CD32/ミドルタワー-型/付属品付)	¥39,800
U8992	V300GP(飯山電機)(PenII(300)/160MB/4GHD/PC8/外装ヤケ/外装汚れ/ミドルタワー型/USB×1/AGP×1)	¥29,800
U8994	M533MS(飯山電機)(Celeron(533)/96MB/8.4GHD/CD40/外装焼け/ミニタワー型/PCI×3/USB×2)	¥39,800

DOS/Vノート

注文No.	商品	価格
U6422	ThinkPad 380(2635-10J)(IBM)(Pen(150)/32MB/1.08GHD/SVGA/AC型番違い/12.1DSTN)	¥19,800
U20075	ThinkPad 380XD(2635-9AJ)(IBM)(MP(233)/32MB/4GHD/CD24/SVGA/擦り傷/12TFT)	¥47,800
U20083	ThinkPad 380D(2635-2AJ)(IBM)(Pen(150)/32MB/1.3GHD/CD/SVGA/外装塗装剥げ・擦り傷/12.1DSTN)	¥23,800
U6234	ThinkPad 560(2640-EJA/EJ9)(IBM)(Pen(120)/24MB/810HD/SVGA/リアゴム無し/PCカバー無し/12.1TFT)	¥14,800
U6381	ThinkPad 560(2640-FJD)(IBM)(Pen(133)/72MB/1.08GHD/SVGA/FD外付け/12.1TFT)	¥19,800
U6558	ThinkPad 560X(2640-65J)(IBM)(MP(200)/64MB/2.1GHD/SVGA/擦り傷/12.1TFT)	¥39,800
U6149	ThinkPad 760EL(9547-J4H/J48)(IBM)(Pen(133)/16MB/1.08GHD/SVGA/12.1TFT)	¥14,800
U6828	ThinkPad iシリーズ(2611-455)(IBM)(MP(300)/64MB/4.8GHD/CD24/XGA/13.3TFT)	¥65,800
U3869	PC-VP13C/WSA1(NEC)(MP(133)/32MB/1.6GHD/SVGA/擦り傷/12.1TFT)	¥14,800
A0114	PC-LB26C62A◆Win98付(NEC)(MP(266)/64MB/4.3GHD/CD24/56k/12.1TFT/付属品付)	¥63,800
A0108	PC-LW30H63D6◆Win98付(NEC)(Celeron(300)/64MB/6.4GHD/CD24/13.3TFT/付属品付)	¥65,800
A0120	PC-LW40H84D6◆Win98付(NEC)(Celeron(400)/64MB/6.4GHD/CD24/56k/14.1TFT/付属品付)	¥89,800
U6863	ARMADA 1590DMT(COMPAQ)(MP(166)/82MB/2.1GHD/CD/SVGA/擦り傷/12.1TFT)	¥37,800
U3921	LATITUDE LM TS30G(DELL)(MP(166)/32MB/2.1GHD/CD10/SVGA/外装擦り傷/12.1TFT)	¥34,800
U6643	ENDEAVOR NT-800(エプソンダイレクト)(PenII(300)/96MB/3.2HD/CD24/XGA/13.3TFT)	¥69,800
U3897	FMV-52NU2S1(富士通)(Pen(120)/16MB/1GHD/CD6/SVGA/擦り傷/11.3TFT)	¥13,800
U6968	FMV-53NU6W1,5(富士通)(Pen(133)/64MB/1.3GHD/SVGA/11.3TFT)……	¥15,800
U3483	FMV-53NP6W5(富士通)(MP(133)/32MB/1.6GHD/CD16/SVGA/外装焼け/12.1TFT)	¥29,800
U6783	FMV-55NA5W5(富士通)(MP(150)/64MB/1.6GHD/CD10/SVGA/イジェクトボタン無し/12.1TFT)	¥29,800
U6461	FMV-5233NP/W(FMV5BNPW5,8)(富士通)(MP(233)/32MB/3.2GHD/CD/SVGA/モデム/ドット抜け/拡張ステーションツメ折れ/12.1TFT/USB×2)	¥42,800
U6859	DynaBook PA1232C9.CW(東芝)(Pen(133)/32MB/2.1GHD/PCカバー無し/擦り傷/12.1TFT)	¥15,800
U6987	DynaBook PAP338J8(東芝)(PenII(400)/128MB/8.1GHD/XGA/PCカバー無し/擦り傷/11.3TFT)	¥76,800
U3560	PA1247C9(東芝)(MP(166)/32MB/2.1GHD/CD/12.1TFT)……………	¥33,800
U20022	PA1262C9(東芝)(MP(200)/96MB/2.1GHD/CD16/SVGA/12.1TFT/USB×1)	¥46,800
U6087	PAP301JC(東芝)(MP(266)/64MB/4.3GHD/SVGA/モデム/リカバリー付き/10.4TFT)	¥62,800
U6577	PAP302JA.B(東芝)(MP(300)/64MB/4.3GHD/SVGA/10.4TFT/USB×1)…	¥72,800
U6520	DynaBook PAS403J.C.B.D(東芝)(Celeron(333)/64MB/4.3GHD/CD24/XGA/14.1TFT)	¥68,800
U6739	PC-5NH01-YA5LA(日立)(MP(166)/32MB/2.1GHD/CD10/SVGA/KB文字剥げ(7以下)/バッテリーパック欠品/12.1TFT/USB×1)	¥26,800
SH789	CF-S51EXJ8◆Win98付(パナソニック)(PenII(266)/128MB/4.3GHD/XGA/11.3TFT/付属品付)	¥59,800
U3794	AL-N2T516J5(パナソニック)(MP(166)/32MB/2.1GHD/SVGA/KB文字剥げ(2)/10.4TFT)	¥34,800
SH785	VAIO PCG-733◆Win98・添付品付(SONY)(MP(200)/64MB/2.1GHD/CD24/12.1TFT/付属品付)	¥69,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

98デスクトップ

注文No.	商品	価格
SH799	PC-9821BP/U7W◆Win3.1付(486DX2(66)/5.6MB/170HD/付属品付)……	¥26,800
SH800	PC-9821BP/U8◆Win3.1付(486DX2(66)/5.6MB/340HD/付属品付)……	¥29,800
U2129	PC-9821XN/U8(Pen(90)/7.6MB/270HD/PCM/2FDD仕様/外装焼け/擦り傷/スロットカバー無し)	¥19,800
SH793	PC-9821V200/SZC2◆Win95付(MP(200)/32MB/3GHD/CD14/付属品付)‥	¥32,800
SH057	PC-9821XA9/C8◆Win3.1付(Pen(90)/8MB/850HD/CD4/PCM/付属品付)‥	¥32,800
U1230	PC-9821XA13/W1P(Pen(133)/16MB/1.2GHD/CD6/PCM)…………	¥11,800
U1636	PC-9821XC16/S5A3,S5B3◆Win95付(Pen(166)/32MB/1.6GHD/CD8/外装焼け/添付品付)‥	¥19,800
U8353	PC-9821XV20/W30(Pen(200)/96MB/3GHD/CD8/64MB増設/外装焼け/擦り傷/PCI×3/ミドルタワー型)	¥42,800
U1433	PC-9821RA223/W30(PenII(233)/32MB/3GHD/CD16/外装焼け)………	¥34,800
SH801	PC-9821RA223/N30◆NT4.0(PenII(233)/64MB/3GHD/CD16/付属品付)‥	¥39,800
U1584	PC-9821RA266/W30(PenII(266)/32MB/3GHD/CD16/外装焼け)………	¥36,800
U8788	PC-9821RA266/W・D30R(PenII(266)/128MB/3GHD/CD16/外装焼け/擦り傷/PCI×2/ISA×3)	¥44,800
U8558	PC-9821RA333/M60(Celelon(333)/32MB/6GHD/CD24/PCIx2)………	¥89,800
U1835	PC-9821F200/X20(MP(200)/32KB/2GHD/CD16/14.1TFT一体型/VGA/外装焼け/PD無し/PCI×1/USB×2)	¥29,800

98ノート

注文No.	商品	価格
U0301	PC-9801NS/R(486SXJ(16)/640KB/外装焼け/ACアダプター無し/モノクロ)……	¥9,800
U3105	PC-9821NE/340(486SX(33)/5.6MB/340HD/拡張バス110pin/9.5TFT)……	¥15,800
U3783	PC-9821NE3/3(増設タイプ)(486DX2(50)/40MB/1GHD/VGA/リアカバー無し・擦り傷/10.1TFT)	¥15,800
U3576	PC-9821LA10/8B(Pen(100)/16MB/810HD/VGA/9.5TFT)…………	¥18,800
U6619	PC-9821NA12/H8(Pen(120)/48MB/810HD/CD4/PCM/SVGA/10.4TFT)…	¥29,800
SH264	PC-9821NA12/S10F◆Win95付(Pen(120)/16MB/1GHD/CD4/PCM/28kモデム/SVGA/12.1TFT/付属品付)	¥32,800
U6319	PC-9821NA12/S10F(Pen(120)/32MB/1GHD/CD4/SVGA/モデム無し/12.1TFT)	¥24,800
U50001	PC-9821NR13/S14Z(Pen(133)/80MB/1.4GHD/SVGA/外装キズ/12.1TFT)	¥32,800
U6393	PC-9821NR15/S10(Pen(150)/24MB/1GHD/CD8/外装塗装剥げ/12.1TFT/8MB増設)	¥33,800
U6075	PC-9821NW150/S20C(MP(150)/32MB/2GHD/CD16/モデム/SVGA/12.1TFT)	¥54,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

MAC

注文No.	商品	価格
SH120	PowerMac 8500/180◆付属品付(PPC604e(180)/32MB/2GHD/CD8/付属品付)	¥25,800
U9472	PowerMac 8500/180(PPC604e(180)/114MB/2GHD/CD8/外装焼け/外装キズ)	¥21,800
U9426	PowerMac 9500/132(PPC604(132)/96MB/2GHD/CD4/外装焼け/擦り傷/汚れ)	¥17,800
U9405	PowerMac 9600/233(PPC604e(233)/64MB/4GHD/CD12/外装焼け/汚れ)	¥29,800
U9378	PowerMac 9600/300(PPC604e(300)/64MB/4GHD/CD24/外装焼け/擦り傷)	¥34,800
U9345	PowerMac G3/MT266(PPCG3(266)/32MB/6GHD/CD24/ZIP・MO付き[未チェックのため保証無し])	¥58,800
U9388	PowerMac G3/400(PPCG3(400)128/64MB/9GHD/CD24/マウス無し)……	¥86,800
U9363	PowerMac 4400/200)(PPC603e(200)/32MB/2GHD/CD12/リアカバー無し/外装焼け/マウス無し)	¥22,800
U9508	PowerMac 8500/132(PPC604(132)/208MB/1.2GHD/CD4/外装焼け)……	¥19,800
U9510	PowerMac G3/DT300(PPCG3(300)/64MB/6GHD/CD/ZIP[未チェックの為保証無し])	¥59,800
U9512	PowerMac G3-350(ブルー)(PPCG3(350)/128MB/6GHD/DVD/擦り傷/UBSマウス無し)	¥73,800

キーボードは別売となります。

PowerBook

注文No.	商品	価格
U3562	PowerBook 190C(68LC040(66)/8MB/500HD/外装キズ/10.4DSTN)……	¥9,800
U3358	PowerBook 550C(68040(66)/12MB/750HD/10.4アクティブカラー)……	¥26,800
U6133	PowerBook 1400CS/117(PPC603e(117)/16MB/770HD/CD6/SVGA/11.3DSTNカラー)	¥34,800
SH786	PowerBook 1400C/133(PPC603e(133)/16MB/1GHD/CD6/漢字Talk7,5,3付き/11.3TFT/付属品付)	¥34,800
U0391	PowerBook 5300CS(PPC603e(100)/32MB/750HD/10.4D-パッシブカラー)	¥13,800
SH796	PowerBook 3400C/240(M5829J/A)◆OS7.6付(PPC603e(240)/48MB/3GHD/CD12/SVGA/12.1TFT/付属品付)	¥57,800
U6992	PowerBook G3(M3553)(PPCG3(250)/160MB/5GHD/CD20/リアカバー無し/汚れ/擦り傷/12.1DSTN)	¥69,800

ディスプレイ

注文No.	商品	価格
U4675	F14T2(NEC)(14.1TFT:1024×768max/外装焼け/3列15pin)…………	¥27,800
U0449	PC-KM153R2(NEC)(15マルチ:1120×768max/0.28mm/外装焼け/3列15pin)	¥7,800
U4826	TFT450(COMPAQ)(14.5TFT液晶:1024×768max/外装焼け/3列15pin)‥	¥27,800
U4561	D1025TM(DELL)(17マルチ:1280×1024max/外装焼け/3列15pin)………	¥9,600
U4818	P990(DELL)(19マルチ:1600×1200max/3列15pin)……………	¥16,800
U4339	FLEXSCAN E55D(ダイヤモンド)(NANAO)(17マルチ:1600×1200max/0.25mm/24.5～92kHz/3列15pin/5BNC)	¥17,800
U4728	LL-T155A(シャープ)(15TFT:1024×768max/外装キズ/3列15pin)………	¥41,800
U4194	CPD-15SF9(SONY)(15マルチ:1280×1024max/0.25mm/外装焼け/3列15pin)	¥7,800
U4277	GDM-17SE2T(SONY)(17マルチ:1280×1024max/3列15pin/5BNC)……	¥14,800
U4411	CPD-20SF2(SONY)(20マルチ:1280×1024max/0.30mm/汚れ/3列15pin)	¥23,800
U4616	RD-17GXII(ダイヤモンド)(三菱)(17マルチ:1280x1024max/0.25mm/外装ヤケ/3列15pin)	¥11,800
U4350	RD-21G(ダイヤモンド)(三菱)(21マルチ:1600×1200max/外装キズ・焼け・汚れ/3列15pin)	¥29,800
U4707	A102G(飯山電機)(17マルチ:1280×1024max/3列15pin)…………	¥11,800
U4335	TXA3601GT(飯山電機)(14.1TFT:1024×768max/フルカラー/外装焼け/3列15pin)	¥29,800
U4881	VS150TW(飯山電機)(15TFT:1024×768max/3列15pin)…………	¥32,800
U4856	CF-L15(パナソニック)(14.5TFT:1024×768max/外装焼け/3列15pin)……	¥24,800

メモリ

注文No.	商品	価格
U7317	ENB-64M(メルコ)(PC-9821NB,LA,NA用:64MB)…………	¥12,800
U8227	NR64-32M(I/O)(PC-9821NR用:32MB)…………	¥5,900
U7626	EMW-32M(16M×2)(メルコ)(ファーストページパリティ有り72pinSIMM:16M×2枚セット)	¥3,680
U81003	VN8-64M(メルコ)(ノート用:64MB)…………	¥8,900

トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

商品の状態により価格の変動がある事がございます。詳しくはTELかネット上でご確認ください。

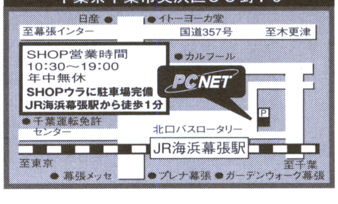
万一、品切れの際はご容赦ください。（12月21日現在）

こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

秋葉原駅前店
千代田区外神田1-15-6 ミツウロコビル4F
TEL.03-5209-6111

大阪日本橋店
浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401

幕張店
千葉県千葉市美浜区ひび野1-9



パシフィック ネット

株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30

計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　ワープロ・ソフトは何をお使いですか？
①Word ②一太郎 ③メモ帳 ④その他

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

① 編集部の学校訪問記…8
② インテル冬のPCスクール…10
③ 大学入試センター試験対策直前マスター…11
④ コンピュータ・ウイルス「BADTRANS.B」撃退法…17
⑤ CPUの歴史と未来をさぐる！…22
⑥ パソコン組み立て実践講座…27
⑦ 特選投稿プログラム・コーナー…35
⑧ 明日のスタープログラマー…79
⑨ Dr.Dのベスト・プログラム研究室…80
⑩ マンガで学ぶ！コンピュータの世界…82
⑪ ブロードバンド通信体験記…86
⑫ 「PIECE」プログラミング…91
⑬ チャレンジ！オリジナルプログラム言語の制作…95
⑭ HSP完全マスター 3Dプログラミング…99
⑮ 図解！Visual Basicゲーム作成術…104
⑯ はじめよう　Microsoft Excelで BASICプログラミング…109
⑰ ナビゲータ搭載の思考型自走ロボット「INSECT」…115
⑱ 21世紀の遊びはコレ！ロボ道場…120
⑲ あくせすROOM／プレゼント…124
⑳ SOFTWARE MAKER HOT INFORMATION…125
㉑ 表紙
㉒ OVER FLOW

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯			応募年月日	年　月　日

キリトリ線

郵便はがき

50円切手をお貼り下さい

1 4 1 - 8 7 1 5

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン　編集部

2月号 行

（フリガナ）		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　　］

	希望番号	品名
ベーマガお楽しみプレゼント②応募欄（124ページ掲載）	番	

キリトリ線

月刊マイコンBASICマガジン② 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄■

※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1		Q2	〔メーカー名〕
Q3		Q4	Q5
Q6			
Q7		〔商品名〕	
Q8			
Q9			
Q10		Q11	〔④の人はソフト名を記入〕
Q12 〔 〕		Q13 〔 〕	
Q14		Q15	

キリトリ線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Windows XP ⑦Macintosh ⑧UNIX ⑨Linux ⑩その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥ゲームボーイ ⑦ゲームボーイカラー ⑧ゲームボーイアドバンス ⑨ワンダースワン ⑩ワンダースワンカラー ⑪その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

ツヴァイ!! Zwei!!

好評発売中

Windows 95/98/Me/2000/XP

DVD-ROM版 CD-ROM版

アクションRPGのファルコムから!

イース、ザナドゥ、ブランディッシュ…etc.数々の名作を輩出してきたファルコムのアクションRPG史に、今、新たなる歴史が刻まれる――。

試してZwei!! キャンペーン
Zwei!!体験版を無料配布中！
詳細はHPで！

■童話の世界をそのままゲーム化したような元気いっぱいのアクションRPG

GAME Watchより抜粋
[Reported by 中村聖司]

impress GAME Watch

結論を先に書くと、万人が楽しめる軽快なアクションゲームに仕上がっていると感じた。ゲーム解像度は640×480ドット固定で、最初のうちは狭っ苦しく感じたが、そのためにふたりのアクションや敵キャラクタたちの愉快な動きを丁寧に見守ることができ、この部分にこそツヴァイ!!の醍醐味が凝縮されている。

グラフィックも2Dベースながら、実に丁寧に描き込まれ、特に攻撃時はもちろん、ことあるごとに画面を彩るエフェクトの数々がゲームを賑やかなものにしている。敵モンスターは、総じて捕獲してペットにしたいぐらい可愛らしく(ペットは1匹別にいるが)、大人がある程度手助けしてやることで子供でも楽しめそうな雰囲気だ。

各ダンジョンは、たんに敵が待ち受けているだけでなく、随所に謎解きやワナが散りばめられており、いきなり敵に包囲されることもしばしばで、とにかく飽きさせないつくりが好印象だ。（中略）

また冒険中に横スクロール式のシューティングゲームも楽しめるという。このあたりにくると、もはや本気なのか冗談なのかよくわからないが、ともかくも本気と冗談の微妙なところをファルコムならではの絶妙なバランス感覚で仕上げたアクションRPGがツヴァイ!!と言えそうだ。

ヤリ込めばヤリ込む程オモシロイ!!

本編のゲーム以外でも、Zwei!!には色んな楽しみが満載！

▼Zwei!!シューティング?! ★隠された×××を見つけだせ！

▼伝説の赤毛の勇者のタイピングや、業界初登りゲー「モナモ～ナ」も遊べるよ！

▼いつでも一緒タマとポチ デスクトップを育てよう！

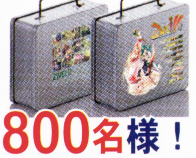
初回限定特典付パッケージの出荷は終了しました。
店頭在庫のみとなりますのでお早めに！

初回限定 四大特典

1. **オリジナル・サウンドトラックZwei!!(Audio CD)**
イースでもない、英雄伝説でもない、ファルコムサウンド新境地！ツヴァイゲームサントラ全曲完全収録！
2. **イース・ヒーリング(Audio CD)**
永遠の旋律達に包まれて揺蕩う忘我の時。イース癒し系完全新作新アレンジ！
3. **永久保存版「Zwei!!」特製シルバーメタリックBOX**
細部にもこだわった銀色に輝く豪華パッケージ。内箱トレイに冒険に役立つ村マップ付き！
4. **ピクチャーレーベル(DVD-ROM/CD-ROM)**
ゲームのDVD-ROMとCD-ROMは、美麗なピクチャーレーベル仕様！

※「オリジナル・サントラZwei!!」と「イース・ヒーリング」の別売も好評発売中！

ブリキ缶CD-BOXプレゼント



毎月抽選で合計800名様！

製品パッケージ内のアンケート葉書をお送りください。詳細はアンケート葉書をご覧ください。

無料プレゼント!! ツヴァイ!! スクリーンセーバー＆壁紙、試聴版「オリジナル・サントラZwei!!＆イース・ヒーリング」

★ゲットするには、ホームページへGo!!
http://www.falcom.com/

Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/

■お求めは取扱い店またはカンタン便利なファルコム通販(falcom.comファルコムドットコム)もご利用下さい。■社員募集(新卒・既卒・中途)！詳しくはホームページ「会社案内・社員募集」をご覧ください。

SONY

愛という感情を、
ロボットに抱くとは
思わなかった。

ココロのAIBO

あなたのもっと近くに。新しいAIBO、誕生。

お問い合わせ

- AIBOインフォデスク 0570-008844
  (ナビダイヤル:全国どこからでも市内通話料でご利用いただけます)
- 携帯電話・PHSから 03-5470-0660
- 受付時間:午前10時～午後6時(年末年始を除く)
- インターネット http://www.aibo.com
  (ネットワークやサーバーなどのメンテナンス期間を除き24時間受付)

● 商品の色は実際とは若干異なる場合があります。● "AIBO"および"AIBO"ロゴは、ソニー(株)の商標または登録商標です。● "メモリースティック"はソニー(株)の商標です。● "ラッテ/LATTE"および"マカロン/MACARON"はソニー(株)の商標です。

- エンターテインメントロボット AIBO "ラッテ"
  ERS-311(写真左) 希望小売価格 98,000円(税別)
- エンターテインメントロボット AIBO "マカロン"
  ERS-312(写真右) 希望小売価格 98,000円(税別)

○ AIBO本体のみでは基本的な動作しかできません。より個性的な動きや性格をAIBOに組み込むためには下記ソフトどちらかが必要です。

- "ラッテ"と"マカロン"それぞれで個性的なキャラクターが楽しめる
  AIBO-ware "メモリースティック"「AIBOフレンド」
  ERF-310AW06J (ERS-311/312用) 希望小売価格 9,000円(税別)
- AIBOを育ててゆく楽しさを味わえる
  AIBO-ware "メモリースティック"「AIBOライフ」
  ERF-310AW01J (ERS-311/312用) 希望小売価格 9,000円(税別)

定価550円 本体524円 雑誌18361-02 T1118361020557

第二十一巻第二号通巻二三六号 昭和五十七年十二月六日第三種郵便物認可 二〇〇二年一月二十五日印刷 二〇〇二年二月一日発行 電波新聞社 〒141-8715 東京都品川区東五反田一―十一―十五 電話〇三(三四四五)六一一一

2002

わかる！動かせる！
プログラムが組める雑誌

マイコンBASIC Magazine

2

誌面刷新
第2号!!

集中連載！大学入試センター試験対策第2弾

数学「BASIC言語」予想問題と解法

http://www.basicmagazine.net/

電波新聞社